Digitized by the Internet Archive
in 2017 with funding from
University of Alberta Libraries

https://archive.org/details/mcfadzean1984

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Robert George McFadzean

TITLE OF PAPER THE DEVELOPMENT OF A MICROCOMPUTER LEARNING PROGRAM APPLIED TO FARM CASHFLOW BUDGETING

DEGREE FOR WHICH PAPER WAS PRESENTED Master of Agriculture

YEAR THIS DEGREE GRANTED Fall 1984

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this paper and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the paper nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) .....R.G.McFadzean.....

PERMANENT ADDRESS:

c/o LAKELAND COLLEGE

VERMILION, ALBERTA

T0B 4M0

DATED 6 JUN 1985

THE UNIVERSITY OF ALBERTA

THE DEVELOPMENT OF A MICROCOMPUTER LEARNING PROGRAM APPLIED TO FARM CASHFLOW BUDGETING

by

Robert George McFadzean

A PAPER

SUBMITTED TO THE DEPARTMENT OF RURAL ECONOMY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Agriculture

IN

Agricultural Economics

Department of Rural Economy

EDMONTON, ALBERTA

Fall 1984

THE UNIVERSITY OF ALBERTA

DEPARTMENT OF RURAL ECONOMY

The undersigned certify that they have read, and recommend to the Department of Rural Economy, for acceptance, a paper entitled THE DEVELOPMENT OF A MICROCOMPUTER LEARNING PROGRAM APPLIED TO FARM CASHFLOW BUDGETING submitted by Robert George McFadzean in partial fulfilment of the requirements for the degree of Master of Agriculture in Agricultural Economics.

.....

Supervisor

.....

.....

Date.....

ABSTRACT

This project involves the use of the microcomputer as a learning tool which may be of considerable use to agricultural students and farm operators in learning important farm business planning principles. Because of declining farm profits due to lower farm product prices and rising input costs, many farm operators in Canada are presently facing financial crises, where they are unable to meet their cash commitments.

The financial analysis and planning tool considered to be most useful to farm managers in avoiding such cashflow crises is the cashflow budget. It provides a systematic way of predicting the cashflow situations arising out of various yields, prices, and costs which may face the farmer, and the probability of his running into cashflow problems leading to the possibility of receivership or bankruptcy. Consequently in this project the subject selected for demonstrating the usefulness of microcomputer programs for learning, was cashflow budgeting.

After a comprehensive discussion of some of the current literature on learning theory and the microcomputer, the project develops a computer program which embodies a number of useful learning characteristics. These are: simulation of real-world situations; individual learner control of path choice, pace, and evaluation; immediate feedback; and the ability to experiment with different values of pertinent variables. The learning program involves the learner in terms of distinguishing the types, sources, and organization of data for a cashflow projection or statement. Exercises are provided to give the learner practice in selecting and organising the appropriate kinds of information for cashflow budgeting. Another exercise presents hypothetical physical and financial farm data from which the learner extracts values to be entered into a cashflow statement. The concluding learning activity challenges the learner to test for and examine the timing and sensitivity effects of changing prices, input costs, yields, etcetera.

There is room for further testing of the cashflow learning program to improve the clarity of what is expected of the user. The program can be expanded to include more possible

situations for cashflow analysis or to include a more complete financial analysis with additional income and balance sheet information.

ACKNOWLEDGEMENTS

The time, effort, and guidance of Alf Peterson as my supervisor in producing this paper is gratefully acknowledged. Committee members Wayne Lamble and Victor Adamowicz provided many helpful suggestions to improve the clarity of the paper and the usefulness of the end product.

As my initial supervisor, Len Bauer deserves credit for basic ideas and encouragement for this project. My association with other faculty members, staff, and fellow students in the Department of Rural Economy has been helpful and stimulating.

I thank my wife, Marit, and my children for their support and encouragement in fufilling the requirements for this degree.

Table of Contents

Chapter Page

I. INTRODUCTION 1

A. Farm Financial Problems 1

B. Financial Statements 1

The Balance Sheet 1

The Income Statement 1

The Cashflow Statement 2

C. Objectives 2

Learning Theory 2

Programming Technique 3

The Organization of This Paper 4

II. THE MICROCOMPUTER AS AN AID TO LEARNING 6

A. Computers and Learning 6

The Basis For Learning 6

The Learning Process 7

The Attributes Of Learning Mechanisms 7

Aids To Learning 10

Conclusions 11

B. The Development of a Computer Learning Package 13

Useful Applications Of Computers In Education 13

The Orientation Of The Learning Package 14

The Learning Environment 15

The Learning Experience 18

Design Considerations & Technique 18

Documentation 22

Testing 22

SUMMARY 23

III. CASHFLOW BUDGETING 24
A. The Cashflow Statement 24
Cash Operating Flows 24
Non-operating Cash Flows 26
The Bottom Line 26
B. Cashflow Analysis 27
Sensitivity Analysis 27
Timing Analysis 27
Long Term Analysis 28
C. Conclusion 28
IV. THE CASHFLOW LEARNING PROGRAM 29
A. Program Charactaristics 29
Memory Requirement 29
Disks 29
Starting Up the Program 29
B. Program Description 30
C. Programming Technique 32
V. SUMMARY AND FURTHER DEVELOPMENT 33
A. Summary 33
B. Further Development 34
BIBLIOGRAPHY 36
APPENDIX A 38
APPENDIX B 45
APPENDIX C 57
APPENDIX D 60

LIST OF FIGURES

Figure		Page
1.	A Learning Matrix	8
2.	The Computer as a Learning Aid	12
3.	The Learning Environment	16
4.	A Cashflow Statement	25
5.	The Warnier-Orr Diagram of the Main Program	39
6.	The Warnier-Orr Diagram of the Farm Financial Flows Figure Presentation	40
7.	The Warnier-Orr Diagram for Either Exercise 1 or 2 in Data Choice	41
8.	The Warnier-Orr Diagram for Data Entry	42
9.	The Warnier-Orr Diagram for Analyses	43
10.	The Warnier-Orr Diagram for the Analyses Options	44

I. INTRODUCTION

A. Farm Financial Problems

In recent years there have been increasing numbers of farm businesses experiencing financial difficulties. High interest rates and decreasing profit margins emphasize the need for more sophisticated farm planning practices. The process of analyzing and planning the operation of a farm business is carried out largely through the use of financial statements.

B. Financial Statements

There are three financial statements which serve to show the farm manager where the business is at, where it is going, and how it will get there. These three statements are the balance sheet, the income statement, and the cashflow statement.

The Balance Sheet

The balance sheet is a financial picture of the business at a given point in time. All assets and their values are listed as well as claims on those assets in the form of liabilities and owner's equity. The balance sheet shows the capital assets and working capital available for production, and the debt burden to be borne by income from operations. When planning farm operations, changes in the balance sheet information can be noted by comparing the balance sheet for the beginning of a period to a projected balance sheet which is prepared for the end of the production period, based on proposed operations during the production period.

The Income Statement

The income statement measures the earning power of the farm business over a given accounting period which is usually a year. From a planning point of view the income budget or projection reveals the estimated profitability of proposed farm operations, which is where the business is going in a financial sense. In order for a business to continue and progress it needs

to be involved in profitable operations. Profitability is indicated by a positive net income, and is necessary at least in the long run if not always in the short run.

The Cashflow Statement

The cashflow statement shows cash inflow and outflow to and from the farm business. From a planning viewpoint, the cashflow budget details the expected flows of funds in a business and shows how the business will operate financially to achieve its projected net income. Cashflow is as important as profitability for financial success, but cashflow is often neglected in analyzing and planning farm operations. Without the financial ability to carry out proposed operations as revealed through cashflow budgeting, the projected profits for those operations cannot be realized. The focus in this paper is on learning to use cashflow budgeting as an important aspect of farm financial management.

C. Objectives

The objectives of this project are to first investigate learning theory, instructional design, and educational computer programming techniques as they relate to the microcomputer as a learning aid. And second to apply this knowledge in the development of a microcomputer learning program. A program suitable for learning cashflow budgeting in a farm financial management course, serves as a vehicle for attaining these objectives.

Learning Theory

Learning is a relatively lasting change in behavior as the result of experience. Learning experiences provide an activity or activities which involve a person with materials in such a way as to bring about learning. The emphasis is on activity and involvement which moves the learner from initial contact with the concepts or skills to be learned, through successive stages of understanding and assimilation, to the point where the concepts or skills can be utilized and applied.

The microcomputer has certain characteristics which can be advantageous in a learning situation. It has the potential to give immediate feedback, learner control (over pace, sequence, and content of the material studied), emulation of real world situations, and the opportunity to experiment with different values of the variables involved.

Programming Technique

The initial design stage is the most critical aspect of the development of an instructional program, while often receiving the least attention. The design of a learning program involves defining the scope, who the user will be, and what learning is to take place. Major objectives and associated learning experiences are developed. The constraints to be put on the learner and the consequences to the learner's actions are specified. Finally the learning experiences are specified to outline what is to be accomplished, learning activities, and standards of performance for self-evaluation.

The use of the microcomputer requires that attention be given to the interaction between the learner and the learning activity. This interaction should involve the learner with the material without requiring too much attention to the workings of the program or the computer hardware. Further attention should be given to using the visual display effectively since this is the major mode of communication between the user and the program.

The logical or instructional design of the program in this project was worked out using Warnier-Orr diagrams (see appendix A) which show what is happening for all modules of the program. Referring to the heading at the left of a particular module in these diagrams tells why it is being carried out. The heading at the right shows how it will be carried out. The vertical sequence tells when a particular module will be carried out. The program design is summarized in five Warnier-Orr diagrams in appendix A.

Once the design has been finalized, the process of translating it into computer code begins. One or more stages of pseudo-code are used where the design is broken into modules and structured like a computer program while still using standard written language for the most

part. Pseudo-code was developed for each module. Specific self-contained procedures are carried out by subroutines which are kept relatively short; less than a page long at the most. This modularization of the program gives good readability and eases the debugging procedure. The program modules are described in appendix B.

From this point the pseudo-code is translated into the programming language code. The IBM BASIC language allows for variable name length up to 40 significant characters and longer as desired. As far as possible this facility was used so that variable names are non-abbreviated English words that give the code good readability. Program variables are listed in appendix C. The final code is structured into sequences, repetitive loops, and alternative routes. The IBM BASIC language has the facility of WHILE-WEND loops, FOR-NEXT loops, and IF-THEN-ELSE conditional branching. These facilities being available, it is possible to avoid unconditional branching which tends to destroy the integrity of a program. An example of the program code is given in appendix D.

Documentation was built up throughout the programming process and consists of the overall design, summary of modules, and comments which are interspersed in the program code. The resulting workable, debugged program should be tested further at several levels of involvement with actual users and the programmer, which could lead to refinement and modification. As presented in this paper, the cashflow learning program has had only a limited amount of testing.

The Organization of This Paper

The paper consists of five chapters. The introduction is outlined in chapter one. Chapter two discusses learning theory, instructional design, the microcomputer as an aid to learning, and the development of a computer learning program. Chapter three covers cashflow budgeting, its importance to farm financial management, the format of the cashflow statement, and the process and scope of cashflow analysis. Chapter four considers the actual computer program with a general overview, overall characteristics, and how the program is used. Chapter

five summarizes the paper and discusses the possibilities of further development of the program and extending the program to other areas of financial planning and analysis. A detailed documentation of the program appears in the appendices including program design, summary of modules, program variables, and an example of the program code.

II. THE MICROCOMPUTER AS AN AID TO LEARNING

This chapter is divided into two sections. The first part, titled *Computers and Learning,* describes how learning takes place, and how various mechanisms contribute to learning with particular emphasis on computer related aspects of these contributions. The second part of the chapter, titled *The Development of a Computer Learning Package,* begins with some of the considerations for using a computer. A framework for an effective design is outlined, followed by a procedure and design considerations for the actual computer learning activities. Guidelines for documenting and testing the final product round out the second part of the chapter.

A. Computers and Learning

The following discussion of the underlying factors involved in learning and the use of computers for learning is based on work by Hudgins (10), Nash & Ball (13), Rockart & Morton (16), and Wydra (19).

The Basis For Learning

There are three major factors which underly the process of learning. First, there must be motivation to learn which is manifested in a focused learning activity. There are theories regarding the absorption of concepts subconsciously, but retrieving concepts "learned" in this way is difficult. Consequently, if the goal is conscious retrieval and use of concepts, then attention to the concepts to be learned is necessary.

The second factor basic to learning is the use of a learning activity. This activity consists of organising relationships between various aspects of what is to be learned, and also between the subject of study and other previously learned facts, skills, or concepts. Such relationships provide the mechanism for retrieval and use of the subject matter. Providing for learning activities is the work of teaching and using learning activities is the work of learning.

A third basic factor to learning is the learner's stage of development. In order for learning to take place, those things to be learned must be placed in relationship to what is

known. Concepts must be in the range of difficulty which is neither lacking in challenge nor beyond the grasp of the learner.

The Learning Process

Assuming the learner is adequately motivated and the subject matter falls within the range relatable to current knowledge, the process of learning centers around organising the new subject matter within a framework of current knowledge.

The achievement of organising the subject matter is largely dependent on the subject matter being meaningful. Concepts of this type will be semantic rather than syntactic; it can be organized in an hierarchal rather than serial fashion; and it need not be memorized.

Rockart and Morton separate the learning process into four areas, beginning with "acquire" where the initial presentation of facts, skills, processes, or concepts takes place. Then through practise and reflection the learner will "embed" the subject matter. This is where the subject matter takes on meaning or is understood by the learner. Thirdly, the learner will "integrate" the new subject matter within existing concepts (current knowledge). Concepts are now in place for the fourth area called "test" where the learner can explore the implications of the new subject matter or how it could apply to new situations. Knowledge expands as the learner continues to move through these four areas of learning - acquire, embed, integrate, and test.

The Attributes Of Learning Mechanisms

In order to investigate where computers could be useful in the learning process, a learning matrix and the ensuing analysis as developed by Rockart and Morton will be used. This matrix is illustrated in Figure 1. The columns of the matrix are the four phases of learning and the rows are the four classifications of the type of learning - facts, skills, established concepts, and frontier concepts. In this context, frontier is distinguished from established in referring to those ideas coming to light through research and experimentation as opposed to ideas which are

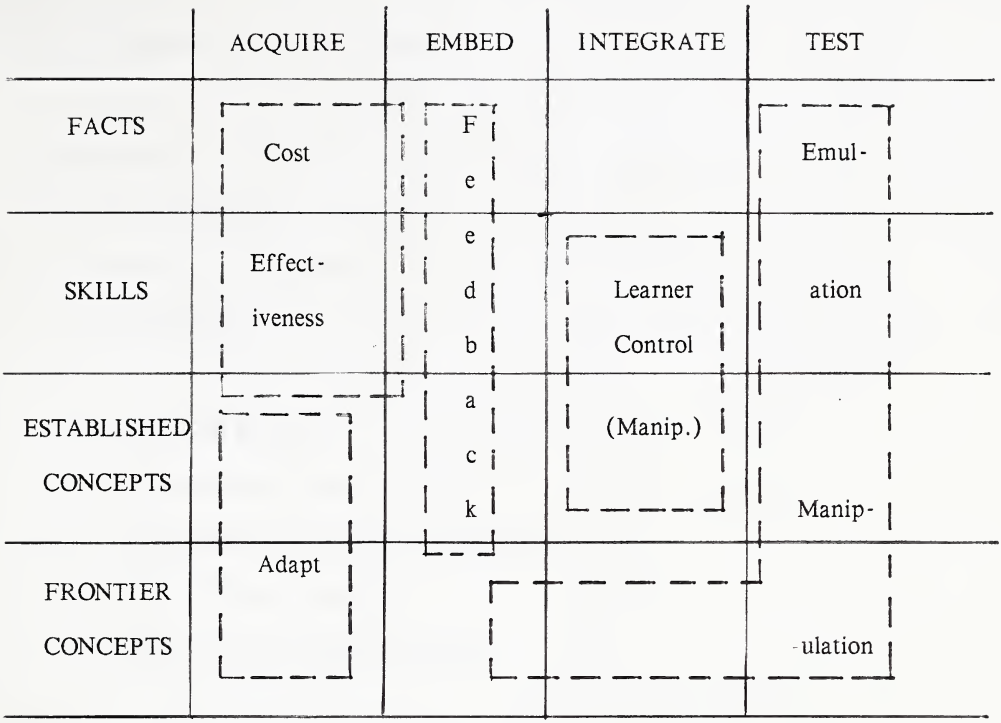

Figure 1. A Learning Matrix.

accepted and in use.

Learning mechanisms are all the possible methods, techniques, devices, etcetera which can be used to bring about learning. A number of attributes of learning mechanisms were analysed by Rockart and Morton and rated as to their applicability in the 16 cells of the learning matrix. The attributes which were investigated were the following:

1. Content Related
    a. Telescope time
    b. Present structure
    c. Rich environment
    d. Provide for ill-structured material
    e. Add new material quickly
    f. Support structured clerical tasks (calculations, formatting)

g. Support unstructured data manipulation

2. User Related
   a. Learner control
   b. Adjust to individual needs
   c. Ease of use

3. Communication Related
   a. Amount of sensory impact
   b. Amount of emotional impact (motivation)
   c. Degree of learner feedback
   d. Access to previous data and concepts

4. Economic
   a. Cost/concept learned
   b. Decentralized availability

In making this analysis, the above attributes were consolidated into the six major attributes of cost effectiveness, adaptability, .feedback, learner control, emulation, and manipulation. The pertinence of these attributes to learning is is illustrated by their position within the matrix. It was found that cost effectiveness was most pertinent to acquiring facts and skills. Adaptability carries more weight when acquiring established and frontier concepts. In embedding or establishing an understanding of facts, skills, and established concepts, learner feedback is the more useful attribute. For the integration of skills and established concepts, learner control is most important with secondary emphasis on manipulation. In the testing of all classes of material and the embedding and integrating of frontier concepts, emulation and manipulation give the best results.

Aids To Learning

Rockart and Morton carried out another analysis with regard to a number of aids to learning to again determine where they were most applicable in the learning matrix. The following is a list of the learning aids investigated:

Drill and Practise(computer)	Class Discussion
Problem Solving(computer)	Laboratory
Paper Progammed Instruction	Videotape
Written Assignments	Lecture
Simulation(computer)	Film
Real World	Television
Tape	Radio
Textbook	Study Modules
Inquiry(computer)	Games(computer)
Tutorial(computer)	

The results of this analysis showed that computers are of most use in the embed, integrate, and test stages of the learning process and of less use at the acquisition stage. The textbook is, and for the foreseeable future will be, the most cost efficient way of acquiring facts. In the area of acquiring concepts where adaptability is of major importance, lecture and classroom discussion are the most suitable mode of aiding learning. At the embed stage of learning, laboratory and computer methods are best in that they are most inducive to giving learner feedback. The usual type of computer programs found at this stage involve drill, practise, and problem solving. There is no learning aid which presently dominates the integrate stage. However, this is an area of potential for the computer where learner controlled instruction can be used. At the test stage of learning, laboratory and computer methods are

again the most suitable aids. Computer simulation and games have been sucessfully used in this area to give the learner an emulation of real world situations with possibilities to explore the outcome of manipulating variables in that situation.

Conclusions

This analysis suggests the profitable use of computers beyond the acquisition stage of learning where the learner is trying to understand, organize, and use the material that has newly been acquired. Figure 2 presents a summary of computer support to learning. This figure correlates the types of computer programs used to aid learning, to the attributes of learning mechanisms and the stages of learning. The computer as a learning aid should attempt to help the learner understand and use newly acquired material by incorporating those attributes which are best suited to the stages of learning beyond acquisition. Computer programs as learning aids should give the learner feedback on what is happening, control over the flow of the program, the possibility of manipulating variables, and real world emulation of the material at hand.

Computers in a learning situation can free learners from tedious calculations and can free teachers from having to assist with all routine matters. The computer will not be able to replace a teacher's ability to share their enthusiasm for a subject, to give encouragement, or to repair lost confidence. Computers can give some appropriate combination of instruction, support, and freedom. They can build confidence by giving immediate feedback to reinforce success or to get back on track. (13)

Learner controlled instruction can be cheaper, more flexible, less aversive to learners and instructors, and have reduced learning time as compared to instructor controlled instruction. Learner controlled instruction can increase the efficiency of instructors by allowing them to take a more specialized, consultive role. Computers accomodate the decentralization of the learning experience. Learning resources and patterns are individualized and learners are allowed to negotiate the content maze in the shortest possible time consistent with their own speed. (19)

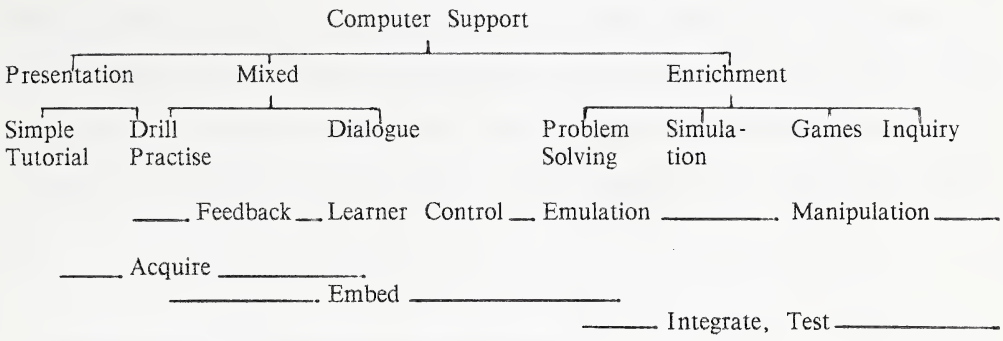

Figure 2. The Computer as a Learning Aid.

In an instruction situation, learners know what they know, the instructor knows the objective, but neither knows what the learner does not know. In instructor controlled instruction, the instructor assumes some level of knowledge for learners, then evaluates and adjusts the assumption. Learner controlled instruction provides the learner with a description of the objectives such that what is not known can be deduced by the learner. The learner is also provided with standards of performance with which to make judgements regarding mastery of the objectives. Having the objectives and achievement standards as well as the subject matter content, an instruction process, and previous learning experience; the learner is then in a position to make an informed decision about pace, sequence, content and mastery in attaining the objectives.

For learner controlled instruction, the instructional effort is invested in design rather than application. An increased demand for software and other materials could shift the emphasis for instructors from regurgitation to the creation of such materials. The design process involves a great deal of analysis and the application of technical knowledge to learning. The key is in the development of measurement tools which allow the specification of objectives and the evaluation of results. Technological advances of today have made available many low cost resources for information and computing capability.

Utilizing the emulation and manipulation attributes of the computer is conducive to the natural way of learning, through making mistakes. Mistakes are an integral part of learning

through deductive reasoning, testing hypotheses, or making estimations. The possiblity of making mistakes in the classroom, on assignments, or on exams can give rise to fears which inhibit learning. A computer program which is patient, helpful, and anonymous will encourage a student to work with the material at hand, to correct mistakes at once, and to move ahead. (13)

B. The Development of a Computer Learning Package

The following discussion draws to a large extent on Wydra (19) who examined learner controlled instruction in general with no specific reference to computers. The learner control aspect of a computer program for learning is considered essential for the effectiveness of such programs as well as being complementary to other important aspects of these programs such as feedback, emulation, and manipulation.

Useful Applications Of Computers In Education

Computers can be useful in cutting costs where the simulated, as opposed to actual process, is cheaper. A program might be used to represent experiences requiring long periods of time such as population studies where the real life experience is not feasible. A dangerous experiment or management of a business can be experienced as a computer simulation, while the real life experience is impractical for students. A computer program is useful where an exact replication or a random choice of a repeatable event is required. (11)

A computer program is not suitable for simple experiments that can be done in the class, for turning pages of print, or for discussion topics. Learner control, which has been identified as a major attribute of computer programs for learning, will be inappropriate if objectives cannot be unambiguously specified or the measurement of progress is not refined to the point where it can be used by a non-expert. Instructional content, learner population, or preparation time may be so limited that it is not worth constructing a program. (1, 11, 19)

The Orientation Of The Learning Package

The first step is to establish the foundation for the learning package by defining its scope, its benefactors, and its benefits.

Define the Mission

The mission of the package is broader in scope than an objective. It outlines what is to be done under which imposed conditions. This definition should not restrict design flexibility during development of the package unless some preconceived limitation is deemed inevitable. (19) As applied in this project, the mission is to be able to set up a cashflow statement with relevant data and use it to analyse the cashflow position of a farm business.

Define the Student Population

Identification of the learner includes specification of probable current skills and knowledge, and probable responses. Analysis can involve the three areas of demography, subculture, and capability. Wydra (19) notes that this analysis is tedious but necessary in order to anticipate responses and the rewards are only visible by the absence of the analysis. He says further:

> "Anticipated obstacles that fail to materialize are imaginary to most, but to the designer they are the awful reality that could have been."

The learners envisioned to be users of the learning program resulting from this project would be students in a farm management course who have had some previous exposure to accounting and financial statements.

Define the Performance Results

Working within the mission and population specifications previously defined, the performance results are a description of the ability to do something under certain conditions and to some degree of expertise or accuracy. These criteria reflect the full breadth of the learning process in going beyond acquisition to the application or utilization

of skills or knowledge. The performance results for the cashflow learning program include the ability to locate and categorize the appropriate data for a cashflow statement; the ability to interpret the information provided by the cashflow statement; and the ability to analyze the effects of changes in production or marketing on cashflow.

Defining the required performance results puts them in contrast to the ability of those using the program and provides guidelines for the construction of the learning activities. The required performance results are also the criteria for measuring the effectiveness of the program.

The Learning Environment

The learning environment is the collection of factors which are pertinent to and necessary for the accomplishment of the mission envisioned. This environment will encompass one or more programs depending on the scope of the mission. Figure 3 shows Wydra's (19) basic idea of the learning environment with only one program illustrated. Moving from a motivation for learning, the program consists of a series of learning experiences, and the package concludes with an assessment which involves all the skills learned.

Objectives

Major learning objectives clearly define the intended outcomes of the programs which are necessary for the accomplishment of the mission. Interrelationships between objectives should be identified. The first major objective of the developed program is to be able to organize relevant information into a cashflow statement. The second major objective is to be able to analyze the effect of changes in production and marketing variables on cashflow. The first objective sets the stage for a correct cashflow statement to be used in the analyses referred to in the second objective.

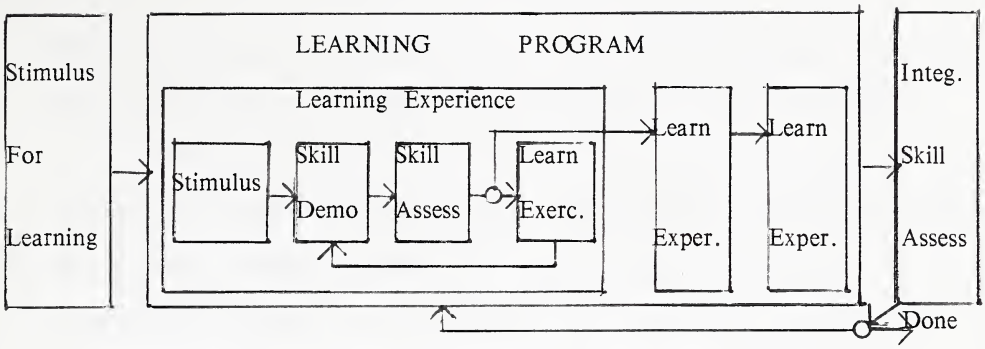

Figure 3. The Learning Environment.

Identify Learning Experiences

Various learning experiences are identified for each objective. Consideration will be given to different types of computer programs which might be used: Simple Tutorial, Drill & Practise, Dialogue, Problem Solving, Simulation, Game, or Inquiry. The cashflow learning program uses a tutorial to relate cashflow to the farm business and two problem solving exercises to complete the first objective. A simulation, with aspects of inquiry and problem solving, is used for the second objective.

Entry Criteria

A review of pre-requisite skills might be provided in the program, in support material, or in teacher presentation. When using learner control design, entry criteria need not be too rigorous since there are alternate paths, remedial loops, and no restrictions on time at points of difficulty. Learner controlled design is used in this project and some previous use of financial statements is assumed.

Decision Delegation

Consideration must be given to the decisions to be given to and the constraints to be put upon the learner. In a learner controlled environment there are decision points, multiple objectives, and multiple learning paths. The design determines the degree to which various paths are made visible to the learner. Any predetermined sequencing should

consider putting easier tasks before more difficult ones, concrete tasks before abstract ones, and providing varying degrees of difficulty to enhance success and challenge.

Pace may be restricted in an instructional package. For a computer program designed for learner control, pacing is by design delegated to the learner. Mode of presentation is evaluated according to the degree to which it can be learner operated, its complexity of operation, and its portability. In consideration of learner control, a computer program can be designed for ease of operation and availability to the learner can be assumed.

Evaluation is usually a matter of substance and procedure rather than form such that there is an objective standard with which to work. Delegation is possible when skills and knowledge can be measured against the standard and interpreted by the learner. Self-evaluation is based on pertinent, timely feedback on performance that can be used to alter sequence, pace, path, and content.

Consequences

Behavior is controlled by consequences so that to predict behavior, it is necessary to predict consequences. A problem can arise when eventual effects are not directly obvious from immediate consequences. In this case there must be either an alternative to the consequences that distort learning, or a counter-feedback to allow the learner to adjust.

Only those consequences which are unique, visible, interpretable by the learner, and directly related to performance can be used. If the consequence is immediate and visible, then it is used as an internal program consequence. For those consequences which may be less visible, it is necessary to build in controls other than those which exist in the normal environment. There must be short term accurate consequences for learner feedback that allows for interpretation of conflicting short-term and long-term effects. Where a void exists, a simulated consequence or feedback mechanism can be contrived. (19)

The learning environment is constructed with cues, feedback, and consequences that focus and accelerate learning. When constructing the learning environment, missing

stimuli and consequences are identified as well as the effect of their absence.(19)

The Learning Experience

The construction of learning experiences synthesizes the research that went into specifying the learning environment into a program that will allow the learner to demonstrate mastery of a skill. A learning experience is constructed for each of the possible stand-alone tasks which fulfill an objective. A task is deemed to stand alone if its performance can be measured in isolation from other tasks. The learning experience elicits a response that can be assessed and provides a mechanism to alter behavior when appropriate. Each experience outlines activities which will result in learning.

The objective of the program is clearly described for the learner including a natural or simulated stimulus for learning. The learners will then know what they should be able to accomplish upon completion of the program. A description of the learning activity gives new information, concepts, and guidance in the form of examples or demonstrations. Performance is assessed so that the learner can self-evaluate mastery and accomplishment. It is a creative challenge to devise adequate measures which are objective and consistent with the expert's rationale rather than subjective. Practise in performing tasks is provided in exercises. A computer program will utilize its advantages of interaction and visual capability. The material used will provide for variety while maintaining relevance to the objective.

Design Considerations & Technique

Under this heading is an accumulation of various specific considerations and techniques which have been found to be relevant to good educational programming. These ideas are taken from a number of the references: Anon (1), Foulis (6), Nash & Ball (13), and Want (18).

Interaction

A rule of thumb regarding computer programs is to have some form of interaction at least every three frames. Learning increases when the learner is involved in making

active responses while studying the material.

The Provision of User Facilities

A program designed for learner control can give the user various options. Options should be continuously displayed or available through a help subroutine. Options are better displayed in a column rather than embedded in a sentence. These options could be to display existing parameters, a menu, or help facilities. The program could provide for a choice between tutorial or command mode, the possibility to anticipate and by-pass certain procedures, and other alternatives. Make termination prior to normal conclusion of the program available. A menu is desirable for a series of programs which are chained together. At the conclusion of a program the choice would be to rerun, display the menu, or end the session. A program facilitates interaction by providing input echo, defaults for null input, suggested answers, and consistency in style for questions and error messages.

Learner control is working when the student is involved in the point in question rather than in a battle with the machine. When anything goes wrong, the program should provide an explanation and escape. An abort or help key should be available as well as the ability to retrace through the material. Giving the learner control of paging avoids frustration with automatic scrolling or scrolling due to the display of error messages.

Proper program design will allow error messages to concentrate on the subject area rather than the mechanics of input. The purpose of feedback is to reinforce learner thinking or redirect their thinking if necessary. Motivating and challenging feedback can take the form of a simple response, a hint, a graphic illustration, or a result due to the input. Well thought through feedback will be able to deal with several right or wrong answers and explain why they are right or wrong.

A heuristic approach through computer simulation should give the learner a realistic involvement in the subject area. Problem solving provides the motivation with

the learner designing the investigation starting at a simple level and developing to a more complex level. Rapid feedback and versatility are key requirements in taking this approach.

The Handling of User Input

Questions are used to develop concepts and are most effective when short, unambiguous, and aimed at the point. The type of response required should be clear so that no guessing is involved. Giving an example of what is wanted is a good tactic for avoiding misunderstanding. Clarity is also augmented by handling both questions and responses consistently. One form of consistency is to require that the return or enter key be pressed after each response. For multiple input, it is best to take each item separately. It is recommended that abbreviations, incomplete sentences, and the coding of answers be avoided with the exception of coding for multiple or long words. Input of various possible types should be anticipated and checked for all possible valid responses. Responses are matched by first letter, key word, or phrase. Allowing more than three tries on input tends to introduce frustration. Consideration must be given to the role of the null input.

Keyword driven interaction is a method of giving learner control. Keywords should be simple and related to the topic under study. Input of keywords could be either by typing them which helps in learning the material or through a menu which is faster as well as avoiding typing errors.

Text & Graphics

Aesthetics of the display will have an effect on the appeal, motivation, and educational value of a program.

Text

The text needs to be legible, easy to read, highlighted, and organized in an attractive manner. Only small portions of text should be presented on the screen with

sentence breaks at natural reading breaks. Attention should be given to the grammar and the use of an appropriate reading level. Variety between adjacent frames is good practise and is achieved utilizing spacing, indentation, and varying display speed. Consideration should be given to the size of print and the use of upper and lower case characters.

There are various techniques for giving emphasis and variety to text such as character size, boxing, inverse, and flashing. Restraint in using these techniques is necessary since over-use would defeat their purpose.

Graphics

Visual images have been shown to be an important part of memory encoding and storage for people. The recall of an image by-passes the process of reconstructing the image from a verbal description. The effectiveness of graphics makes it especially important that they accurately represent the real world to avoid misconceptions. It is also important to be aware of the costs in technical expertise, programming time, and computer memory requirement for producing graphics. A learner may call into question the cost of a graphic by dispensing with it in a matter of seconds.

A successful graphic might be a central display with information used throughout the program; it might give integrity to a disjointed or complex subject which is expanded upon in supporting material; it might be an illustration of an essentially visual kind of learning.

A graphic display should be organised in a manner consistent with its message and not beyond the degree of detail necessary. Movement and animation may enhance the display when they are pertinent to the objective. It may be necessary to subdivide complex illustrations. It may be necessary to provide perceptual guidelines pointing out the relevance of the image. The level of complexity must be within the learner's grasp and it must be balanced between the boring and the novel.

Documentation

Useful documentation of three types can be provided to give general information, an introduction to users, and a program description. Documentation of general information could include title, author, language, machine, memory use, subject areas, level, running time, special requirements, brief synopsis of aims, and content. Introductory instructions to teachers and students would give a brief and clear outline of the information necessary to use the program. This information would include a description of the overall concept of the program. Constraints and freedoms should be pointed out including which decisions the learner will make and which decisions are to be or have already been made by others. The learner should have an understanding of how to proceed, any special situations that may arise, and how to seek help. To aid users in planning, the inclusion of an estimate of average time to negotiate the package is helpful. A program description would start from a broad outline and move through successive levels of increasing detail to the program listing. (5, 6, 19)

Testing

Wydra(19) recommends the testing of all underlying assumptions involved in specifying population, required performance results, environment, objectives, learning experiences, and entry criteria. Espich & Williams(5) suggest three levels of testing with revisions made after each test. First there would be several sessions of programmer-learner working with the program to note areas of confusion, inconsistency, total stumbling blocks, or an indication that the program is talking down to the learner. The ease and effectiveness of branching through the program due to interaction should be noted. A second level of testing would involve a small group of learners who have had no prior contact with the programmer. The learner would be advised of the trial basis of the program and be asked to note difficulties. Pre and post tests for learner capability are given and following the trial a discussion between learners and the programmer is held. The third level of testing is to have the program put to use with no indication that it is being tested.

Analysis can be carried out on a quantitative basis from the scores on the learner capablility tests as an indication of the level of achieving the objectives of the program. Other analysis could include time used, ranges of scores, or most missed frames.

SUMMARY

The development of a learning package envisioned here starts with the definition of what is to be accomplished, by who, how they will get there, and when they will have arrived. With this basis, specific learning activities are developed and worked into a functional, effective package through documentation and testing.

III. CASHFLOW BUDGETING

The farm manager is mainly interested in two financial aspects of proposed farm operations. These are the profitability of those operations and the financial ability of the farm business to carry through those operations. It is this latter ability of the farm business to meet its financial obligations which is the subject of cashflow budgeting.

A. The Cashflow Statement

The cashflow statement gives an accounting of the movement of cash in and out of a business over a period of time. These flows of cash arise from farm operations in the form of cash operating revenues and expenses, and from non-operating sources and uses of funds. These flows in and out of the business' cash assets or working capital are detailed in the cashflow statement. In addition, the cashflow statement shows the surplus/deficit position of these cash flows, and the fluctuations in accumulated operating loan and working capital over time. See figure 4 for the types of items included in a cashflow statement.(3, 17)

Cash Operating Flows

Cash operating flows arise from the production and marketing activities of a farm business. In making projections for operating flows, historical data, past experience, and possible effects of weather, disease, and insects are important considerations.(2)

The farm business purchases supplies and services for cash, and utilizes these together with its capital assets for production. In the budgeting process, the kinds, amounts, prices, and timing of purchases of supplies and services to be used are projected in detail. These projections form the basis for the dollar amounts included in the cash operating expense section of the cashflow statement.(2, 3)

Marketing of produce results in cash receipts which flow into the business' working capital. Here again, the budgeting process requires detailed projections. Crop yields, crop grades, numbers weaned, numbers of finished animals, finished weights, product prices, and

CASHFLOW STATEMENT

For the Period 01 Jan 1984 to 31 Dec 1984

	1st Qtr	2nd Qtr	3rd Qtr	4th Qtr
Non-operating Inflow				
Capital Sales	0	0	0	0
Term Credit	47500	0	0	0
Borrowed				
Contributions	0	0	0	0
Investments	0	0	0	0
Liquidated				
Total	47500	0	0	0
Non-operating Outflows				
Capital Purchases	50000	0	0	0
Term Credit Repaid	0	0	0	26000
Withdrawals	1500	1500	1500	1500
Investments Made	0	0	0	0
Total	51500	1500	1500	27500
Cash Operating Revenue				
Crop	0	0	55272	0
Livestock	0	1250	0	55396
Custom Work	0	0	0	0
Term Interest	0	0	6	157
Total	0	1250	55278	55553
Cash Operating Expense				
Crop	1024	11924	4061	0
Livestock	14925	3650	4165	5550
Supplies & Services	294	1665	4570	11230
Oper. Loan Interest	315	900	585	0
Total	16558	18139	13381	16780
SURPLUS/DEFICIT	-20558	-18389	40441	11273
Begin. Cash Balance	500	942	553	915
Term Deposit Change	0	0	500	11500
Operating Loan Change	21000	18000	-39000	0
End. Cash Balance	942	553	915	688
ACCUM. TERM DEPOSIT	0	0	500	12000
ACCUM. OPERAT. LOAN	21000	39000	0	0

Figure 4. A Cashflow Statement.

timing of sales are the kinds of details which form the basis of the dollar amounts included in

the cash operating revenues section of the cashflow statement.

Non-operating Cash Flows

Sources and uses of funds, which are not associated directly with operating activities, are designated as non-operating cash flows. The farm business acquires and alters its resources of capital assets through the use of non-operating cash. This will include, for example, the purchase and sale of land, machinery, and buildings. Borrowing will probably also be involved along with principle payments on loans. The farm business is regarded as a separate entity from the owner's personal financial affairs. The finances of owners and their farms are connected through owner's contributions and withdrawals for living expenses or other personal uses. These uses and sources of personal funds for the farm owner are non-operating cash flows for the farm business. The farm as a business entity may also make off-farm investments or liquidate such investments.(2, 3)

The Bottom Line

The bottom line of the cashflow statement is the cash balance position of the farm business at a particular point in time. The usual format of a short term cashflow statement covers the period of one year in monthly intervals such that the cash balance is shown at each month end. The surplus/deficit position at the end of each month is the sum of all operating and non-operating cash inflows less cash outflows during the month. The cash balance at the end of each month is the cash balance less the accumulated operating loan at the beginning of the month plus the surplus/deficit for the month. In that a negative cash balance is not feasible, it is covered by and shown as accumulated operating loan requirements. (12)

In actual practise the cash balance is maintained above some minimum level by drawing further operating funds when a deficit cash flow draws working capital below the set minimum. The minimum cash balance works as a safety net when the amounts and timing of cash flows vary from that which was projected. On the other hand, when working capital rises above some

predetermined level it will be placed in a term deposit. With an arrangement of this kind, a positive cash balance shows as accumulated term deposit and a negative cash balance shows as accumulated operating loan.

B. Cashflow Analysis

The analysis of cashflow focuses on the ending periodic cash balances throughout the time period as reflected in accumulated operating loan and term deposit. The accumulated operating loan cannot exceed the farmer's credit limit and should normally be paid down to near zero between production periods. These criteria are monitored by the farm manager during the analysis process as modifications are made to the cashflow statement. The modifications can be considered in terms of when and how much the flows of cash are changed. These two facets of cashflow analysis are considered separately as sensitivity and timing analysis although they will often be interrelated.

Sensitivity Analysis

Sensitivity analysis involves the identification of ranges in operating projections which can be sustained before the cashflow position is likely to become untenable. Some of the operating projections which could be investigated in this manner are a possible drop in crop yields, a rise in input prices, a drop in product prices, a drop in animals marketed due to disease, or a rise in interest rates.(2, 3, 12, 15)

Timing Analysis

The timing of purchases and sales can have a pronounced effect on cashflow. Both operating and non-operating flows are subject to this type of analysis and often different prices are associated with different times. The price of fertilizer may be less earlier in the year than at seeding time, but the earlier outflow of cash will incur an additional interest expense and must be sustained by the business' working capital for an additional length of time. A farm business

might be able to maintain operations within its operating credit limit by selling grain off-board at an earlier time rather than waiting for delivery quotas. Borrowing opportunities can be investigated to note the effects of different timing of payments and pay back periods.(12, 15)

Long Term Analysis

Long term cashflow analysis is basically the same process as has been discussed for the short term except that the period of time considered is extended to three, five, or more years with yearly intervals rather than monthly intervals. The emphasis shifts from seasonal to cumulative, far reaching, or sporadic effects of some changes to cash flow.(12)

Many of the kinds of changes discussed under sensitivity analysis and timing analysis could be investigated for their cumulative effect over a number of years. It might be useful to note the ability of the farm business to withstand the long term effect of a depressed product price. Inflation may work its way into the cashflow statement as increased interest charges, input prices, or product prices. An effect of inflation that might show up is proportional decreases in fixed loan payments as other amounts are increasing with inflation.(12)

A major change in scale or type of operations will have far reaching effects which can only be fully investigated over a long term. Never-the-less, the short term effects of such a change will also be important to note.

Many non-operating cash flows occur at non-uniform times over a period of years. Some thought in predicting and experimenting with these sporadic sources and uses of funds may reveal some advantage to a particular timing of these flows. This investigation may also reveal a potential difficulty along with an early start in seeking a solution to the problem.

C. Conclusion

Cashflow budgeting is as useful as the detail and attention which goes into it. If correct and realistic data forms the basis of the statement entries and some thought is given to the possibilities in the future, then the cashflow analyses can lead to sound financial decisions.

IV. THE CASHFLOW LEARNING PROGRAM

A. Program Charactaristics

Memory Requirement

The program is divided into sub-programs which are loaded and run sequencially except when the choice to review causes a branch back to a previous sub-program. The division into sub-programs allows a random access memory capacity of 64K bytes to be adequate.(7)

Disks

The disk operating system (DOS) and the BASIC language interpreter are on the program disk. An auto executing sequence is also provided to initiate the learning program. The user can proceed with no other disk, but when available a disk in drive B will be used to record the level at which the user is working when a session is terminated. At the next session the program can then return to the level left by the user. The other use of the data disk is for the user to retain files of any cashflow statement assembled in the section on liquidity analysis.

Starting Up the Program

The learning program is initiated by simply inserting the program disk in the left hand disk drive "A" and turning on the machine. The first sub-program is automatically made available. When the user chooses to conclude that first segment, the next appropriate sub-program is brought into use. From this point on the sequence of sub-programs used will depend on the user's choice to conclude or review a part of the program.

B. Program Description

The program is divided into two major sections that coincide with the two aspects of the learning objective. The first part helps the learner relate data sources to appropriate cashflow data and how to organize it into a useful format. The second part gives the user an example of the origin of the numeric entries and practice in using the cashflow statement for analysis of the farm business.

The first section called "Data Choice", involves the user in the type of data which is pertinent to cashflow budgeting. This section is again broken down into two parts, the first of which is called "Farm Financial Flows Figure". An illustration is built up step by step on the screen to show how the flows of funds affect the cash assets of a farm business. A running commentary accompanies the growing figure which concludes by relating the cashflows to the cashflow statement. The presentation stops at numerous points and waits for direction before continuing. At these halt points, it is also possible to choose to start the presentation from the beginning again, to conclude the presentation, or to conclude the session with the computer program.

The choice to conclude the farm financial flows figure presentation moves the program on to the second part of the data choice section. "Exercises", as this second part is called, consists of a demonstration and two exercises. The demonstration shows how to use specific keys and procedures in the exercises. Exercise 1 involves the user in distinguishing cash inflows from cash outflows. Exercise 2 makes a further distinction between operating and non-operating flows. In both exercises a random list of data is presented. These items can be re-ordered in appropriate categories or deleted if deemed irrelevant to cashflow.

A number of options are available through the function keys. A "help" screen displays the use of various keys and the procedures for moving or deleting items. The farm financial flows presentation can be reviewed or the completed figure alone can be displayed on the screen. The user can choose to conclude the current part of the program whereupon the next part is presented. An exercise can be restored to its initial state or the exercise can be redone

with a new set of data. Learners can choose to check their work whereupon their ordering of the data items is placed next to the correct order such that discrepencies are apparent. There is also the opportunity to call up comments on why a specific item is placed in a given category. The data is presented in the form of a cashflow statement with numeric data and it is possible to have the statement calculations done. The numeric data and calculations are not essential to these exercises, but are included to set the data labels in proper perspective to a cashflow statement.

The second major section of the program is called "Liquidity Analysis". This section is further divided into two parts where a cashflow budget is set up and then analyzed.

"Data Entry" presents a cashflow statement devoid of numeric data and a description of a farm. The objective is to extract pertinent data from the farm description and enter it into the appropriate cells of the cashflow statement. A cursor is moved around to indicate the current cell. Certain keys are used to page through the various categories of the cashflow statement and the pages of the farm description. Pressing a numeric key initiates numeric entry to the current cell.

Options available through the function keys include a help screen showing the use of various keys, and the opportunity to review "Data Choice". The data which is entered into the cashflow statement can be saved on or retrieved from the disk. The "redo" option clears the cashflow statement for a new start. The program can be directed to make a paper printout of the cashflow statement. The evaluation places the user entries beside the correct entries for each quarter of the year. User entries are highlighted when they differ from the correct entries. Ending the session returns the computer to the control of the disk operating system.

The "Analyses" part of the program presents a number of possible analyses that can be carried out using the cashlow budget which was set up in "Data Entry". Having chosen one of the analysis options, the objective is displayed and then any additional data or instructions on how to proceed is given. The control available to the user and options available through function keys is the same as in "Data Entry" with a few exceptions. A summary page and

calculation are additional features. The evaluation focuses on the summary page and also presents comments on these calculated results. It is possible to check the changes in entered data as well.

C. Programming Technique

Computer programming has some intrinsic goals in addition to meeting the objective of the design. These are efficiency in development and accessability to modification and upgrading. This can be accomplished by approaching the program as a whole from the beginning and breaking it down into modules of ever increasing detail until the level of computer code is reached. Re-working the program is kept to a minimum by maintaining from the start the logic which achieves the design objective. The modular structure gives clarity and readability to the program such that it is relatively easy to find any given segment and know what that segment is doing.(4, 8, 9, 14)

Each self-standing subroutine is no longer than one page in length and will be structured in the form of a sequence, a repetition, or an alternation. The program code is formatted with headings for the subroutines and indentation to highlight the structure. Understanding is further enhanced by using non-abbreviated variable names which are indicative of function. Too many comments within the program code can be distracting and should be used only when necessary.(9)

V. SUMMARY AND FURTHER DEVELOPMENT

A. Summary

This paper presents a theoretical base and methodology for the development of a microcomputer program as a learning aid. The use of the microcomputer's potential to provide quick feedback, learner control, emulation of the real world, and manipulation of paramenters can involve the learner with the material in an individualized and unintimidating manner that can be especially conducive to learning.

Emphasis is placed upon using an extensive and thorough design stage at the beginning of the development of a learning program. The design moves from the initial broad concept through successive stages of increasing specification and detail. The program is eventually broken down into sub-procedures which are described using normal written language in computer program-like format. This pseudo-code may also go through successive stages of increasing detail until the final translation yields the program code, although only one stage was used in this project. Development of the program in this manner produces documentation in the form of the specified design and pseudo-code. The final stage of program development was to test the working, debugged program through various degrees of programmer proximity to the program in use, but further testing is required.

Cashflow budgeting was presented as an application for a microcomputer learning program. Cashflow budgeting involves one of the three major financial statements used in farm financial management. The other two described were the balance sheet and the net income statement. Because efficient use of working capital has been critical in recent years of high interest rates and ever decreasing profit margins for farm businesses, cashflow budgeting was considered to be an important aspect of farm financial planning which farm managers should learn to use, and therefore a useful subject for a computer learning program.

The microcomputer program was developed on the basis of the learning theory and development methodology presented in chapter 2, with the subject of cashflow budgeting as

presented in chapter 3. The first of two major sections of the program was designed to help in learning the origin of cashflow data and how it is organised in a cashflow statement. A graphic illustration and exercises at two levels of difficulty were activities used to involve the learner in this section of the program.

The second section of the program proceeded from the concepts presented in the first section but switched emphasis from the types of data pertinent to cashflow to the sources and entry of numeric data. In addition this second section went on to involve the learner in various kinds of analysis of the cashflow statement.

The program was designed to delegate control of pace and evaluation to the user. The evaluations were organised to give the learners a standard of performance by which to self-evaluate their work. Other options made available in the program were help screens that explain the availability and use of keys and procedures, the choice to start a segment of the program over again, the opportunity to save and retrieve data on disk, or to print the cashflow statement. The intention of incorporating the forgoing features in the program was to give the users a learning experience where they could work with the material in stages, move on when they felt comfortable, and not have to be concerned about the workings of the program or computer machinery.

B. Further Development

There is potential for expansion of the "Analyses" part of the program. More options could be offered including input of interest rates which are presently fixed in the program at 13% for the operating loan and at 10% for the term deposit. Other variations in the analysis options would allow for even further expansion of this segment of the program.

Further development of the program could also move in the direction of including the projection and analysis of profitability, change in net worth, and financial ratio analysis which accompany the various cash flow budgets. The cashflow statement is the logical beginning point in a more total financial analysis since it needs the least amount of data. The net income

statement for example, needs the non-cash revenues and expenses, depreciation, and inventory data in addition to cash flows. The balance sheet needs data on beginning values of fixed assets, term liabilities, and owner's equity in addition to cashflow and net income data. The connection between this additional information, the net income statement, and the balance sheet could be illustrated with expanded versions of the farm financial flows figure. Data entry would be expanded to include the additional information and statements. The analysis of proposed farm operations and modifications to those proposals could then include looking at net income, changes in owner's equity, debt/equity and other financial ratios, and returns to assets and equity.

Another development envisioned would be a program where the learner could enter basic parameters like prices, amounts, yields, finished animals, depreciation rates, interest rates, beginning levels of assets, inventories, and liabilities, etc. This data would be collected for an existing farm operation or a proposed beginning farm operation. The program could then process the entries and calculations for preparation of the financial statements. A number of suggested sensitivity and time related analyses plus any others the learner deemed appropriate would form the basis for a report on the financial well being of the farm operation for the projected year. An exercise of this type would empahsize the connection between the realistic, basic data being used and the results shown up in the financial statements, as well as dealing with the analyses themselves.

One final development can be mentioned in connection with the program outlined in the previous paragraph. This program could be expanded to allow for the possibility of long term analysis. The time period would be extended from the second year for 3 to 5 years, with yearly intervals. It would then be possible to analyze cumulative and sporadic effects as well as major changes in the scale or type of farm operation.

BIBLIOGRAPHY

1. Anon. *A Guide To Developing Instructional Software for the Apple II Computer.* St. Paul, Minnesota. Minnesota Educational Computing Consortium. 1980.

2. Anon. *Micro Farm Accountant - Cash Budgeting.* Columbia, Missouri. Agricomp, Inc. 1984.

3. Barry, P.J., Hopkins, J.A., and Baker, C.B. *Financial Management in Agriculture.* Danville, Illinois. The Interstate Printers and Publishers, Inc. 1979.

4. Booth, T.L. and Chien, Y.T. *Computing Fundamentals and Applications.* Santa Barbara, Calif. Hamilton Publishing Company. 1974.

5. Espich, J.E. and Williams, B. *Developing Programmed Instrucional Materials.* Belmont, Calif. Fearon Publishers, Inc. 1967.

6. Foulis, M.B. "Guideline For The Evaluation Of Micro-based CAL Applications". *Involving Micros In Education.* R. Lewis, E.D. Tagg, editors. New York, New York. North-Holland Publishing Company. 1982.

7. Graham, L.J. *Your IBM PC.* Berkeley, Calif. Osborne/McGraw-Hill. 1983.

8. Graham, Neil. *Programming the IBM Personal Computer: Basic.* New York, New York. Holt, Rinehart, and Winston, Inc. 1982

9. Higgins, David E. *Program Design and Construction.* Englewood Cliffs, New Jersey. Prentice-Hall, Inc. 1979.

10. Hudgins, B.B. *Learning and Thinking.* Itasca, Illinois. F.E. Peacock Publishers, Inc. 1977.

11. Kosel, M. *Designing Instuctional Computing Materials.* St. Paul, Minnesota. Minnesota Educational Computing Consortium. 1981.

12. Mumey, G.A. *Optimal Farm Debt.* Farm Light and Power. Regina, Saskatchewan. Farm Light and Power Publications Ltd. 1982.

13. Nash, A. and Ball, D. *An Introduction To Microcomputers In Teaching.* Florence, Kentucky. Hutchinson & Co. (Publishers) Ltd. 1982.

14. Prather, Ronald E. *Problem Solving Principles With Pascal.* Englewood Cliffs, New Jersey. Prentice-Hall, Inc. 1982.

15. Rehman, T. and Esslemont, R.J. *Computers In Farming- Millstone or Milestone?.* Reading, England. Farm Management Unit, University of Reading. 1981.

16. Rockart, J.F., and Morton, M.S. *Computers and the Learning Process in Higher Education.* New York, New York. McGraw-Hill Book Co. 1975.

17. Shillinglaw, G., Gordon, M.J., and Ronen, J. *Accounting - A Management Approach.* Homewood, Illinois. Richard D. Irwin, Inc. 1979.

18. Want, D.L. "Keyword Driven Interaction In Computer Assisted Learning". *Involving*

Micros In Education. R. Lewis, E.D. Tagg, editors. New York, New York. North-Holland Publishing Company. 1982

19. Wydra, F.T. *Learner Controlled Instruction.* Englewood Cliffs, New Jersey. The Instructional Design Library, Vol. 26. Educational Technolgy Publications, Inc. 1980.

APPENDIX A

Design Outline

The design of the program is summarized in the following five figures which outline the program as Warnier-Orr diagrams:

Figure 5 - The Main Program.

Figure 6 - The Farm Financial Flows Figure.

Figure 7 - Exercise 1 or 2in Data Choice.

Figure 8 - Data Entry

Figure 9 - Analyses

Figure 10 - Analysis Option

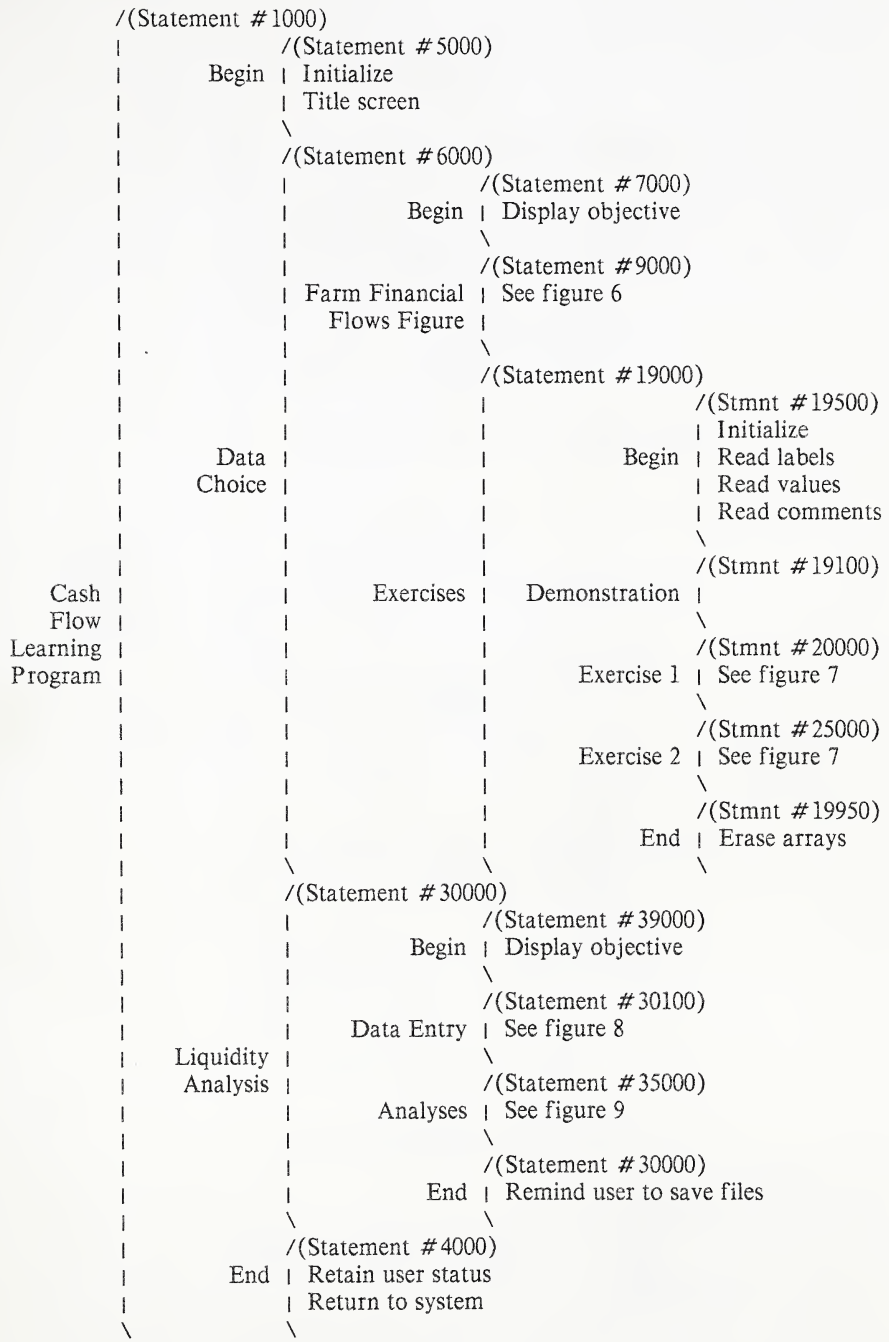

Figure 5. Warnier-Orr Diagram of the Main Program.

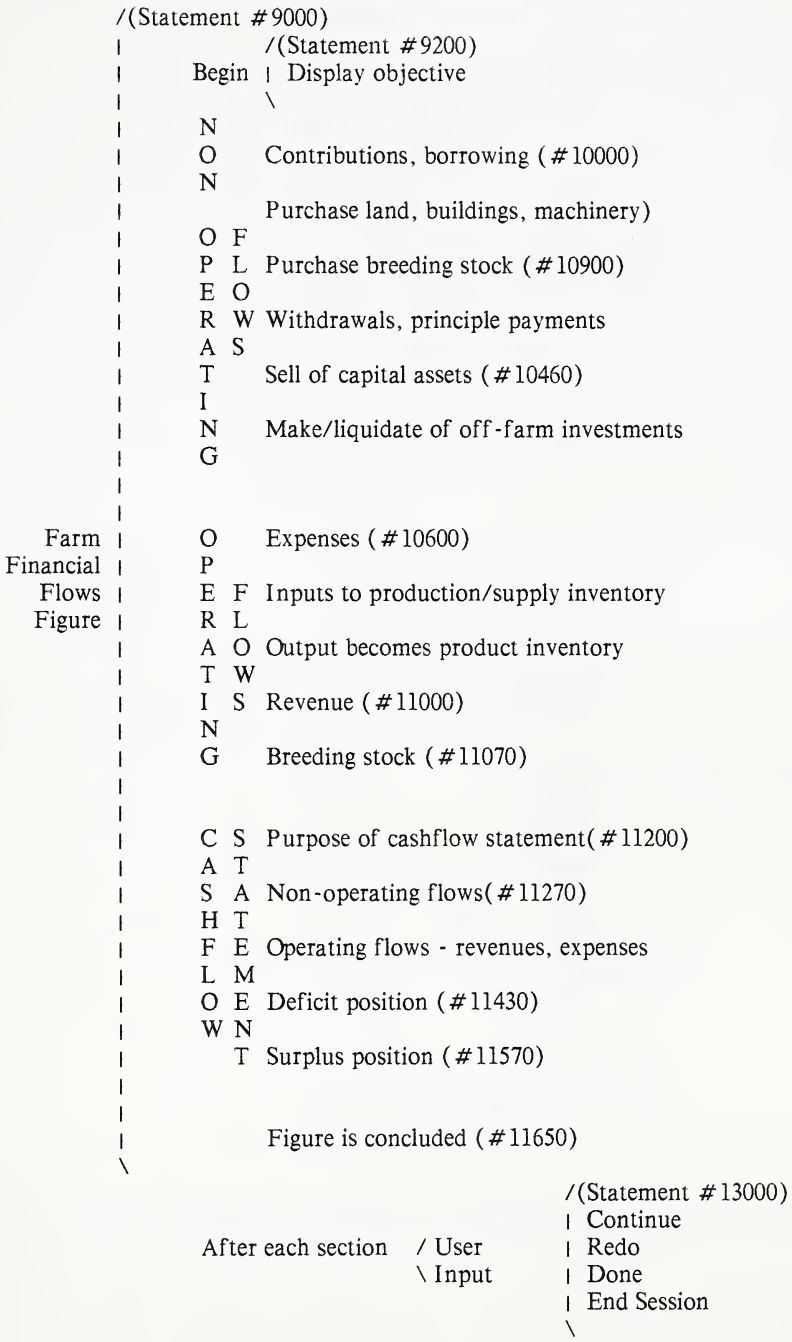

Figure 6. Warnier-Orr Diagram for Farm Financial Flows Figure Presentation

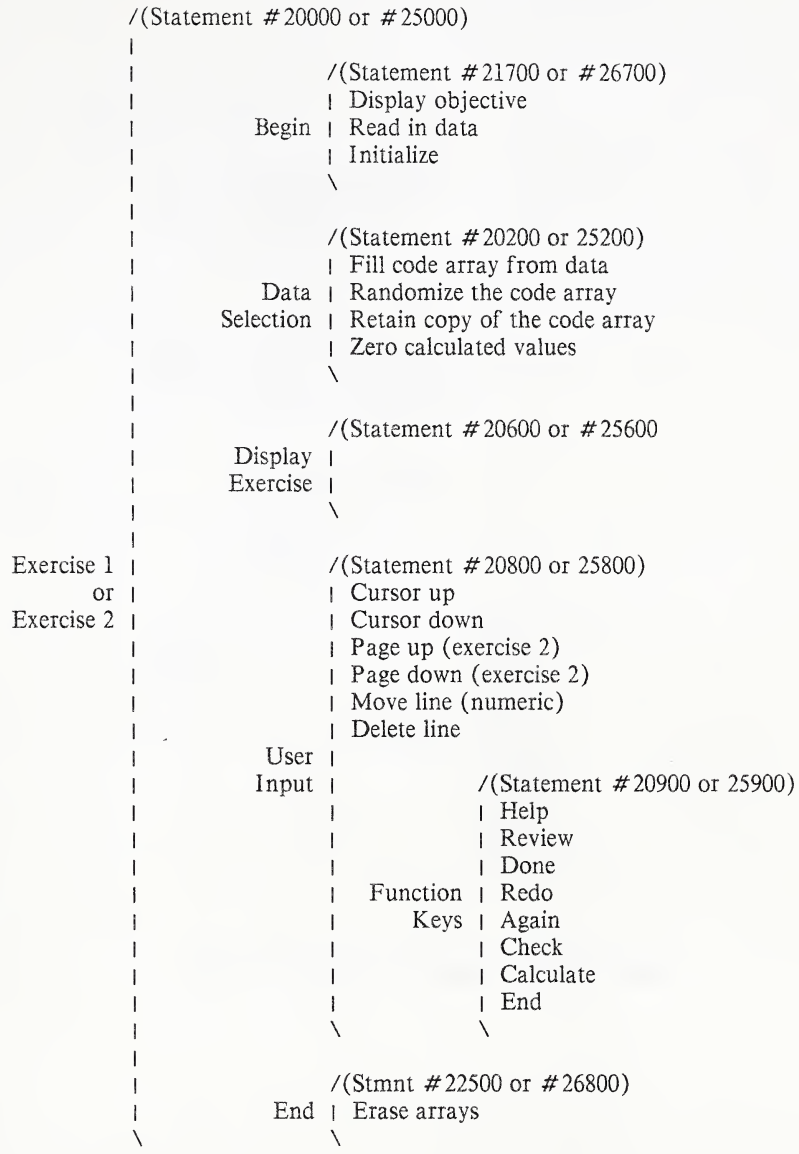

Figure 7. Warnier-Orr Diagram for Either Exercise 1 or 2 in Data Choice.

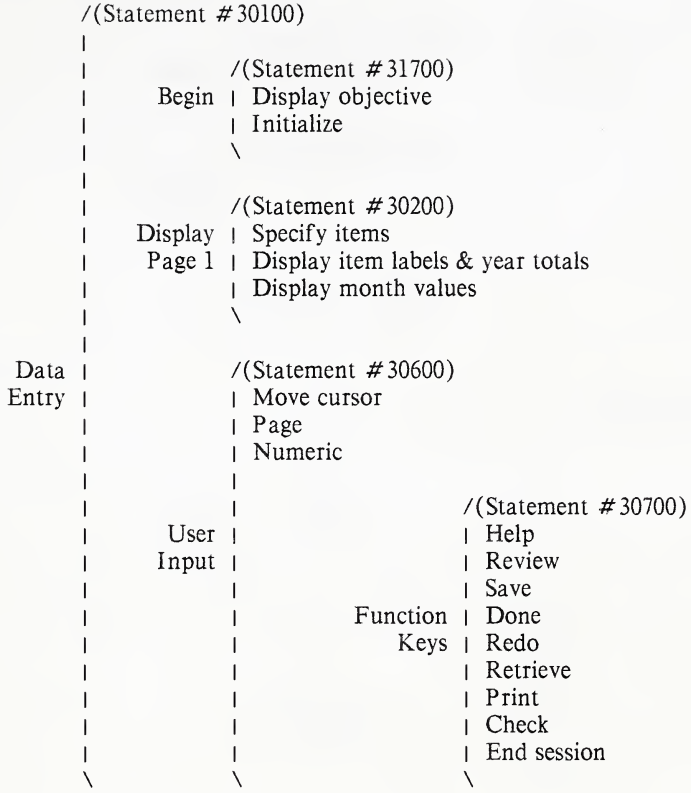

Figure 8. Warnier-Orr Diagram for Data Entry.

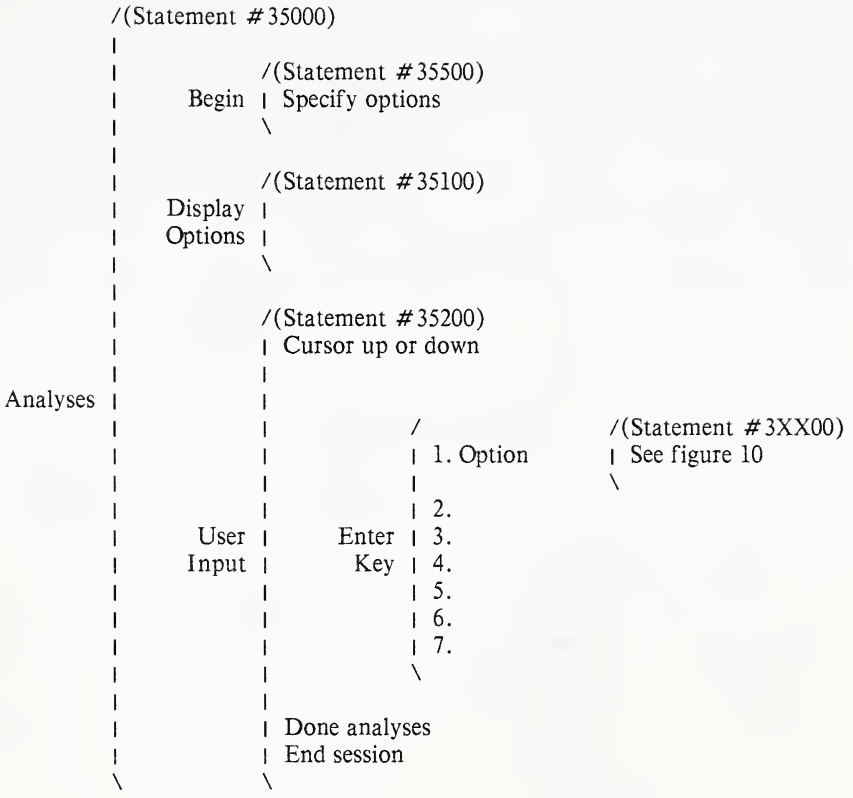

Figure 9. Warnier-Orr Diagram for Analyses.

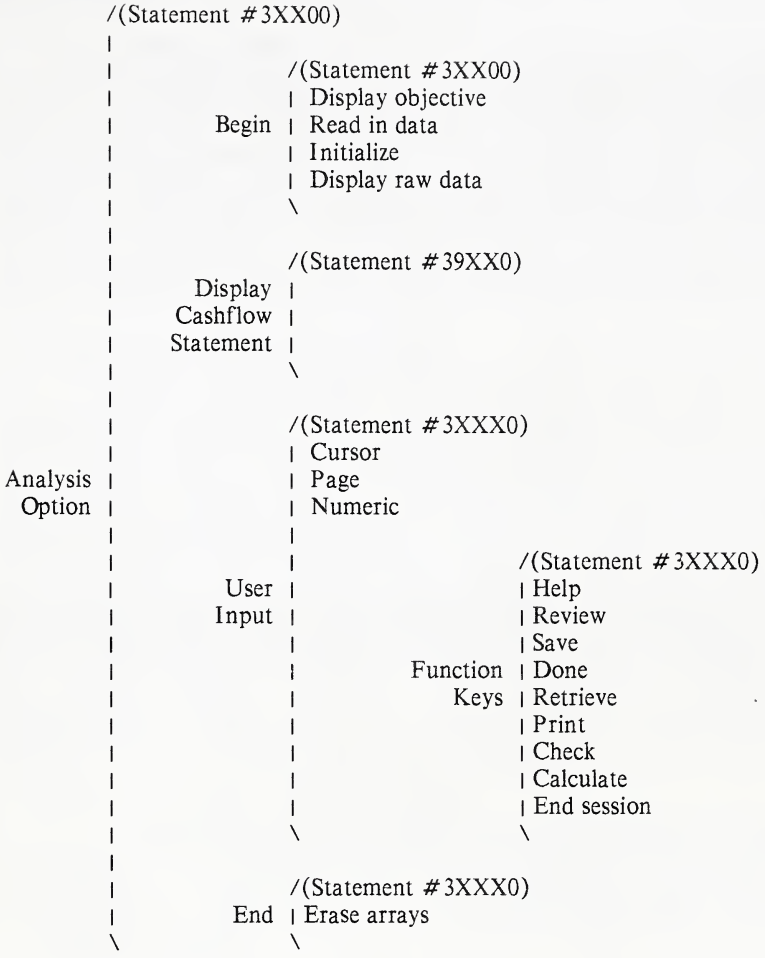

Figure 10. Warnier-Orr Diagram for the Analyses Options.

APPENDIX B

Summary of Modules

Cashflow Learning Program

The computer program begins with some basic initialization and displays the title screen where cashflow is defined and its importance is noted. The program is divided into two major portions. The first portion, called "Data Choice", deals with the issue of being able to bring appropriate data into cashflow budgeting. The second portion, called "Liquidity Analysis", involves cashflow budgeting proper where the student is given practise in entering data into a cashflow statement and analyzing the calculated results. In conclusion, the program records a variable called status which indicates the level at which the user is working when the option to conclude the session is taken. At the subsequent session, the program uses the status value to return the user to the same level again.

Data Choice - Farm Financial Flows Figure

The objective of the farm financial flows figure is to illustrate schematically the flows of funds in a farm business and the origin of cashflow data. The step by step presentation of the figure with comments can only execute sequentially. However, at any of the numerous points where keyboard input is required in order to continue, the user is given the opportunity to choose three other options. Pressing function key, F4 ends the presentation of the flows figure; pressing function key, F5 starts the presentation at the beginning again; and pressing function key, F10 ends the session with the program as a whole. These three options are controlled through the variables DONE.FLOWS, REDO, and DONE.ALL. The execution of each segment of the flows figure presentation is dependant upon these three variables remaining false, as is their initial condition. If any of the three appropriate function keys are pressed, then the corresponding variable is set to true. All remaining segments of the presentation are by-passed to return to the beginning of a WHILE-WEND loop within which the presentation executes. If either of DONE.FLOWS or DONE.ALL is true, the loop is exited. If REDO is true, then it is reset to false and the presentation starts from the beginning.

Farm Financial Flows Figure Presentation

The farm financial flows figure presentation consists of the display of various labelled boxes with interconnecting arrows, highlighting, and comments. The step-wise segments of the presentation can be divided into three parts dealing with non-operating flows, operating flows, and the relationship of those flows to the cashflow statement.

Non-operating Flows

Owner contributions and borrowing are presented as sources of funds to the farm business. Regardless of how funds are held, they are designated as a cash asset. The farm business with cash assets in place can convert those cash assets to productive capital assets like machinery, buildings, and land. Breeding stock is also presented as a capital asset such that the purchase or sale of breeding stock would be a non-operating flow. Since all livestock sales are generally regarded as operating flows, a difficulty arises which is addressed at the conclusion of the following part on operating flows. Withdrawals and principle payments on loans are other non-operating flows which draw on cash. Selling capital assets brings funds into the farm business. Making or liquidating off-farm investments decreases or increases cash assets.

Operating Flows

Capital assets are used in the productive operations of the farm business. Financial flows associated with these production activities are operating flows. Expenses are incurred by purchasing supplies in the required amounts and at the given prices. Where expenses are being projected for budgeting purposes, the amounts and prices of supplies must first be projected. Most inputs go directly to production, but any left over are accounted for in supply inventory. As production proceeds, output becomes product inventory. Information on crop yields and numbers weaned is needed to project levels of output. Produce is marketed to bring in revenue with cash receipts flowing into cash assets.

Consistency in dealing with breeding stock requires a choice between breeding stock as a current asset with purchase and sells as operating flows; or breeding stock as a capital asset with purchases and sells as non-operating flows; or the use of an intermediary transaction to move breeding stock from a capital asset to current asset before selling. Breeding stock purchases and sells are regarded as non-operating flows here. The choice of breeding stock as a current asset with purchases and sells as operating flows is as likely, if not more likely to be encountered.

The Cashflow Statement

The cashflow statement accounts for the movement of cash in and out of the farm business over a period of time. All non-operating flows either increase or decrease cash. Cash operating revenues increase cash, while cash operating expenses decrease cash. When cash outflows exceed inflows, term deposits are redeemed to cover the deficit which decreases cash. If there is still a deficit, then it is covered by an operating loan. When cash inflows exceed outflows, any accumulated operating loan is paid off as far as possible. If there is still a surplus, then cash is increased with a term deposit.

Data Choice - Exercises

The data choice exercises are set up with lines of data in random order, the challenge being to rearrange the lines into a useful ordering.

Demonstration

The methods of manipulating the lines to bring them into order is demonstrated. Lines are numbered and a cursor can be moved up and down in front of the lines to point to a particular line. A given line is moved by moving the cursor to it and keying in the line number to which it is to be moved. Lines can also be deleted to remove irrelevant information. The facility to have the calculations done is also demonstrated. Although the numeric (quarterly) data and the calculation facility is not necessary to the exercises, it is included to give a feel for a complete cashflow statement. During the demonstration, user input is restricted to the appropriate key as indicated on the screen. The program will simply "beep" and continue to wait if other keys are pressed.

Exercise 1

In exercise 1 the randomized list of cashflow data is to be ordered into the two categories of cash inflows and cash outflows.

Data Selection

There are eighty data items from which ten are selected each time the exercise is started. Each item is identified by a code and the selection consists of an array with fifteen codes. The five items, in addition to the ten selected items, are the calculated lines: Total Cash Inflow, Total Cash Outflow, Cash In Less

Cash Out, Operating Loan, and Cash Balance. The latter four of these calculated lines are initialized as and remain as the last four items in the selection array.

The selection routine always sets the first two items as Total Cash Inflow and Operating Loan Interest, and then sets the selection array pointer to 3. The random choices are made by a subroutine given the number of choices to be made, and the range of codes identifying a category. One choice is made from non-operating inflows, two from revenues, and one from non-operating outflows. The next two choices are for crop and livestock expenses and are matched to the revenue choices. Two choices are made from supply and service expenses and one choice from irrelevant items. A subroutine is called to randomize the first eleven items in the selection array. A copy of the selection array is made to be used if the user chooses to redo the exercise, and in doing the evaluation. All calculated values are initialized to zero.

Choosing Routine

This routine works within a loop which is exited when the number of choices to be made reaches zero. Consequently, when a valid choice has been made, the number of choices to be made is decremented and the selection array pointer is incremented. The random choice is made in the given range and then tested against six items not to be used. If a match is found, the rest of the loop is by-passed and the choice is redone. A nested loop checks the current choice against previous choices in the current range. If a match is found, the rest of the loop is by-passed and the choice is redone in order to avoid duplicates.

Randomize a List

Working from the beginning to the end of the list, a random choice is made in the range from the next item to the last item. This random choice is then exchanged with the current item.

Display Exercise 1

The exercise title is displayed at the top, the available function keys at the bottom, and the cashflow statement header on line three or lower depending on the variable EMPTY which indicates the number of lines which have been deleted. A loop displays the lines corresponding to codes in the selection array using three formats. Most lines are displayed with a line number, item label, four quarterly values, and a year total value. Total Cash Outflow and Cash In Less Cash Out lines lack a line number since they are not available to be moved. Operating Loan and Cash Balance lack a line number and a year total value. Double spacing is used after the totals for inflows and outflows, and after Cash In Less Cash Out. Finally, variables to indicate the current item and line number are set to one and the cursor is displayed at the beginning of line one.

User Input

A variable, which indicates that the user is done with exercise 1, is set to false as well as another variable indicating that a valid key was pressed. Reading of the keyboard and consequent action takes place within a loop which is exited when the user chooses to end exercise 1 by pressing function key, F4 or end the session by pressing function key, F10.

A subroutine, which is universal to the program, reads the keyboard and sets an appropriate variable to true when a given key or key within a given range of keys is pressed. These variables have all been previously initialized to false.

If an appropriate variable is found to be true, then a subroutine is called to carry out the corresponding action, a valid key is indicated, and the keyboard reading variable is reset to false. Often the exercise must be re-displayed as well.

If no valid key was pressed then a message to indicate this is displayed.

Cursor Up

The current position of the cursor is blanked out. Both the item number and line number are decremented. If this brings the item number to zero, then item and line number are reset for the bottom line. If the line for Total Cash Inflows is encountered, the line number is decremented again because of the double space. The cursor is re-displayed at the new line.

Cursor Down

This subroutine works on the same principle as Cursor Up with the item and line numbers being incremented rather than decremented.

Delete a Line

Neither the calculated lines nor the Operating Loan Interest line can be deleted. The cursor is fixed from its usual blinking state, and a message indicating the item to be deleted is displayed. The user can abort the deletion by pressing the escape key or allow it to proceed by pressing the enter key.

The current selection array code is set to zero, and then moved to the top of the selection array. The variable indicating empty lines is incremented. Note that the ITEM variable indicates both the current item in the selection array and the line number when there are no deleted lines. When there are deleted lines, ITEM indicates the line number, but ITEM+EMPTY indicates the corresponding item in the selection array.

Move a Line

A numeric key pressed indicates that a line is to be moved. The cursor is fixed and a message indicates the line to be moved and the number of the key pressed as the target line for the move. Another message indicates that only the backspace, number, enter, and escape keys will be recognized. Another input routine deals with this specific situation. Pressing escape aborts the move. Backspace and number keys are used to modify or add to the target line number. Pressing enter indicates that the move should go ahead. A loop exchanges the current item code with the next item code in the appropriate direction until the target line is reached.

Help

The help screen has been loaded on disk and is loaded directly to the screen memory. This display indicates the cursor keys, explains the function keys, explains the procedure to move a line, and explains the procedure to delete a line.

Review

The cursor is fixed and a line at the bottom of the screen indicates three previous parts of the program which can be reviewed: Completed Financial Flows Figure, Flows Figure with comments, and the objective. A cursor is displayed in front of the first choice and a loop checks user input. The cursor can be moved right and left, and escape aborts the review command. Pressing the enter key indicates that the cursor is at the desired choice and the input loop is exited. The appropriate subroutine is then invoked to carry out the desired review.

Redo

The selection array is restored to its original state using the copy which was made at the conclusion of the selection process. Calculated values are reset to zero and EMPTY (lines) is reset to zero.

Again

The choice to do the exercise again invokes the Data Selection routine for a new choice of data.

Print

The screen characters are sent to the printer after the modification any non-standard characters and the insertion of carraige returns.

Evaluation

The screen is cleared and a new title line displayed. A header indicates a column on the left for "YOUR CHOICE" and in the center "CORRECT CHOICE". On the right a box is displayed where comments come on the screen. A message indicates that the evaluation is being set up. Three subroutines make the correct ordering, set up the evaluation array of item codes, and display the evaluation. A line is displayed at the bottom to indicate that three function keys are operable: F4 for done with the evaluation, F7 to print the screen, and F10 for done with this session. Control passes to an input loop which allows the user to move the cursor up and down, page up and down if necessary, register the enter key to display a comment on the item pointed to by the cursor, or register one of the function keys.

Correct Order

The evaluation array is cleared and the original selection array is read into the bottom of the evaluation array. The range of codes for cash inflows is specified and a subroutine, called Category Search, moves any of the selected codes in that range to the beginning of the untouched selected codes. The same procedure is used for the Total Cash Inflows line, the range of outflows codes, and the Total Cash Outflows line.

Set Up Evaluation

The summary line of the first category is indentified as Total Cash Inflows and a series of subroutines carries out a sorting procedure as follows. First, all of the items in the user's selection for the given category are included in the final evaluation array. For each of these items, a search is made of the correct order and if a match is found, it is so marked and the item is removed from the correct order. Secondly, all unmatched items remaining in the correct order are added to the final evaluation array and marked as unmatched. Third, the summary line is added to the final evaluation array and marked as matched. The same procedure is followed for the second category of items. If the final length of codes becomes too long for a single page, then the two categories are divided between two pages.

Display the Evaluation

This routine is common to both exercise 1 and exercise 2. In exercise 1 the display will take one or two pages depending on the number of mismatched items. In exercise 2, up to four pages may be needed. The range of items to be displayed is determined according to the page number and number of items in each category as noted in setting up the evaluation.

For the determined range of items, each item is displayed under one or the other, or both of "YOUR CHOICE" and "CORRECT CHOICE", depending on how they were marked during the set up procedure. When a summary line is encountered, there is a double space. The variable identifying the current item is set to the first item accept when the first displayed category is empty as may happen in exercise 2. When this exception occurs the current item is set to the

second item. Finally, the cursor is displayed.

Select Comment

The cursor is fixed. The current item is matched to the appropriate comment and a display subroutine called.

Display Comment

This routine is common to both exercises and is mostly involved with dividing the comment into short lines without breaking a word. A loop within which the dividing process and display of the short lines takes place is exited when the first character of the next portion is found to be blank or empty. From the maximum of 21 characters dictated by the comment box, a second loop shortens the line one character at a time until the last character is blank or empty, or the next after last character is blank. The acceptable line is then displayed and variables governing the line to be displayed are rearranged for the next portion of the comment. When the user proceeds from the pause routine, the comment is erased and the cursor is restored to a blinking state.

Calculation

The cursor is fixed and a message indicates that the calculation is in progress. The year totals are initialized to zero. An outer loop steps through the four quarters. The category totals for the current quarter are set to zero. An inner loop adds the value of each item in the first category for the current quarter to the category total for the current quarter, and also to the year total for each item. When this loop is done, the category total is added to its year total. A second inner loop carries out the same procedure as the first inner loop, but for the second category and includes the calculation of operating loan interest. The surplus/deficit value is set equal to Total Cash Inflows less Total Cash Outflows and is then added to its year total.

For the first quarter, beginning cash balnce and accumulated operating loan are assumed to be zero. Cash balance is set equal to beginning cash balance plus surplus/deficit less accumulated operating loan. If cash balance is negative, then operating loan is set equal to the positive value of the negative cash balance, and cash balance is set to zero. Otherwise, cash balance remains at its positive value and operating loan is set to zero.

Exercise 2

Exercise 2 is an extended version of exercise 1 using twenty randomly chosen items rather than ten. These items are to be ordered into four categories on two pages with a summary of the cashflow statement on a third page. Due to the similarity in procedure, only the differences in exercise 2 are noted and reference should be made to the exercise 1 documentation for the basic procedure.

Data Seclection

Eight choices are taken from non-operating and operating inflows, nine choices from non-operating and operating outflows, and one choice from irrelevant items. After randomizing the list, the lines for Total Non-operating Inflows and Cash Operating Revenue are moved to their places at the bottom of page one.

Display Exercise 2

This routine is used for pages one and two and depending on the page, the appropriate range of items and the appropriate part of the header, are specified. Only one format is used to display all lines. Separate variables are used to keep

track of the item number and the line number. There are also separate variables for each page to keep track of empty lines.

Display the Summary Page

This routine specifies the line number for each item and calls a subroutine that displays the line with one of two formats with or without the year total and inserts double spaces where needed.

User Input

User input in exercise 2 includes the facility to page up and down, and calls different subroutines.

Move a Line

In exercise 2 this routine includes two extra lines to detect moves from one page to another and calls one of four subroutines to make preparatory moves in these cases.

Move Page 1 to 2 with Empty Lines

The item is moved to the top of page 1 and exchanged with the last empty line on page 2. Variables indicating empty lines on both pages are adjusted.

Move Page 1 to 2 without Empty Lines

The item is moved to position 10 and then exchanged with the first item on page 2 while avoiding moving the Non-operating Outflows line to page 1.

Move Page 2 to 1 with Empty Lines

The item is moved to the top of page 2 and exchanged with the first line on page 1. Variables indicating empty lines on both pages are adjusted.

Move Page 2 to 1 without Empty Lines

The item is moved to the top line of page 2 and exchanged with the next to last item on page 1 while avoiding moving the Non-operating Inflows line to page 2.

Help

The help display includes additional information on paging keys.

Review

The review command includes the option to return to exercise 1.

Evaluation

This is the same procedure as in exercise 1, but includes extra operations to deal with the extra categories and extra pages.

Calculation

Following the summations for each of the four categories, the lines for Total Inflows, Total Outflows, and Surplus/Deficit are obtained and added to their year totals. Cash balance is placed into accumulated term deposit assuming a minimum cash balance of zero.

Liquidity Analysis - Data Entry

In this second major division of the program the item labels are given in ordered, categorized pages, and the emphasis switches to the entry of numeric data and the analysis of the cashflow statement. The first section, called Data Entry, involves the entry of

monthly data for a total farm budget for one year. The second section, called Analyses, involves the analysis of various modifications to the cashflow situation presented in the first section.

For Data Entry the cashflow statement is presented on sixteen pages, eight pages for each of the first and the last six months of a year. Each page shows one of various categories of non-operating flows, revenues, or expenses. Only monthly values are shown and no summary page is used in this section. Also displayed on each page is information on the farm operation for which the budget is being made. This raw data can be paged through separately from the cashflow pages.

Set Up Display

For each category, the heading, the range of items, and screen lines to be used are specified. A subroutine is called to display the item labels.

Display Item Labels

The screen lines between title and information box are cleared. The heading is displayed followed by the item labels for the given range.

Display Month Values

Depending on which six months are to be displayed, the appropriate month headings are displayed. The six month values for each item are displayed for the given range of items.

User Input

The variable DONE.DATA.ENTRY is set to false. See User Input under Data Choice - exercise 1 for the basic working of this routine. The facility to move the cursor and page in four directions is available. Numeric input is used to enter values in the cashflow statement. There is also an extra paging facility for the raw farm data.

Cursor Up or Down

The cursor is blanked out at the user input routine. If the variable designating the screen row falls below the first item row then row is re-specified to the last item row. If row rises above the last item row, then the row is re-specified to the first item row. The variable DISPLAY is set to true and the cursor display routine is called.

Cursor Right or Left

When screen column positions for the monthly values fall out of range, they are re-specified to the other end of the monthly values. The cursor is displayed.

Erase/Display Cursor

When the variable DISPLAY is false, this routine converts the entry at the cursor to non-inverse by inserting the code 7 in the second byte of the screen memory for each character. When the variable DISPLAY is true, besides specifying the code 112 for inverse display, the variables MON and ITEM are determined to identify the current month value at the cursor.

Paging

Paging subroutines re-specify the page when it falls out of range and call the appropriate subroutine to re-display both item labels and monthly values or simply the second set of monthly values when paging right or left.

Numeric Input

On line 23 the words "Numeric entry:" plus six spaces for the maximum length of number are displayed. The input loop allows for escape, backspace, cursor

right, cursor left, number, and the enter keys. Pressing enter exits the input loop. Only valid number characters are included in the entry and it is displayed at the cursor in inverse. Finally, the display on line 23 is erased and the current monthly value is set to the entered value.

Help

A screen display is loaded off disk and explains the various keys for cursor moves, paging, number entry, and the available function keys.

Review

Review offers the option to return to Data Choice. The user data is saved and the Data Choice part of the program is loaded into memory.

Save

The user data can be loaded onto disk for use at a subsequent session.

Done

Data Entry is concluded and the program proceeds to Analyses.

Redo

Redo clears all the monthly values. The user is warned that this action will nullify all previously input values and is given the opportunity to abort this command.

Retrieve

User files which have been previously saved on disk are displayed and the user can choose to have any one of these files retrieved and displayed on the cashflow statement.

Print

The raw data and cashflow statement are printed.

Evaluate

The evaluation display is similar to the entry display, the difference being in the display of monthly values. Monthly values are displayed in quarters with the user entries to the immediate right of the item labels and the correct values to the far right. the user can page through all categories, all four quarters, and also page through the raw data to check their work.

End

The user is given the option to return to Data Entry in case they wanted to save a file or inadvertantly pressed the F10 key. The go ahead to end the session returns control to the disk operating system.

Liquidity Analysis - Analyses

The user is presented with a series of possible cashflow analyses from which to choose. The format for all of these analyses will be the same with only the challenge presented, some additional raw data, and the instructor's comments on results being different for each analysis.

Display Options

A series of liquidity analysis problems are displayed in a column and a cursor is displayed in front of the first option with the variable CHOICE set to 1.

User Input

The input loop is exited when the user presses function keys F4 for done liquidity analysis or F10 for done this session. The cursor can be moved up and down in front of the various options. On pressing enter, control branches to the appropriate option routine.

Display Cashflow Statement Set Up

The cashflow statement is displayed on 14 pages using the same categories as for Data Entry except that livestock and other revenues are on one page, crop and livestock expenses are on one page, and a cashflow summary is included. Totals for each category are displayed and total cash operating revenue and expense are displayed with revenue and expense categories. The year total for each item is also displayed accept for accumulated term deposit and operating loan. For each page the first screen line, category heading(s), item range(s), first item in part 2 if necessary, and "extra" total lines are specified. The routine for displaying item labels and year totals is then called.

Display Item Labels and Year Totals

The first category label heading and heading for year totals is displayed. A loop displays the items and is exited when the last item code is exceeded. If the first item in part 2 is encountered, an extra space is inserted and the second heading is displayed. An extra space is inserted before total lines. On the summary page, spaces are inserted before five of the lines for appropriate spacing. The item label is displayed and if the item is not accumulated term deposit, accumulated operating loan, or minimum cash balance, then the year total is also displayed. After exiting the loop for printing the item labels, an additional line may be displayed as designated by the variable SUB. These additional lines are the totals for non-operating flows and cash operating flows which are taken from the cashflow summary sequence of items.

Display Month Values

Depending on which six months of the year is to be displayed, the appropriate header for these months is used. A loop displays the month values using the same cirteria for spacing as outlined for item labels. The month values for a SUB line are also displayed as necessary. The cursor is displayed on the first row of items for all pages except the summary page which consists of entirely calculated values.

User Input

The same format for user input as has been used previously is maintained in this section. Full cursor movement, paging, numeric input, and all function keys are available.

Review

The options are to display the challenge presented for the given problem or to return to Data Choice.

Redo

Monthly values are reset to the correct values of the original cashflow situation which was arrived at in Data Entry and was the beginning position for the present problem.

Print

The cashflow statement is printed.

Evaluate

The summary page is divided into four sets of items which can be displayed both for the first six months and the last six months. The correct results of a set of items are displayed at the top of the screen with the user results underneath. At the bottom of the screen there is a comment box with the instructor's comment on the results. The user can page through the four sets of items and also call data entry check to evaluate changes to the input data.

User Input - Analyses Evaluation

The input loop allows the user to page through the four sets of items of the summmary page and is exited on pressing function keys F4 when done with the evaluation or F10 when done the session. Function key F5 also gives excess to check data entry.

Display Analyses Evaluation - Labels & Year Values

The lines for function keys and title are displayed. The first item to be displayed is calculated according to the set of items being displayed. Headings for correct results, year, and user results are displayed. A loop displays the three items of a set placing the item label and correct year value at the top and repeating the item label, but using the user year value underneath with the user year value being highlighted if it is different from the correct value.

Dispaly Analyses Evaluation - Month Values

The first item is calculated, first screen line is set, and the cursor is fixed. The header appropriate to the months is displayed. An outer loop processes the three items to be displayed and an inner loop processes the appropriate six months to be displayed. If there is a discrepency between the correct and user month value, then the user value is highlighted.

Display Analyses Evaluation - Comment

The comment which corresponds to the set of items and six months being displayed is selected. The display area is cleared and then the current comment is displayed. The cursor at the lower, right of the screen is set to blinking to indicate readiness for user input.

Calculation

The cursor is fixed and a message indicating that calculation is in progress is displayed. All year values are set to zero and the first month interest on the term deposit is calculated. A loop runs through each of the twelve months. For other than the first month, the interest on accumulated term deposit and accumulated operating loan of the previous month is calculated. For each category of items, the summation line and range of items is specified. A subroutine is called to do the summation for each category. Also included in this loop are the calculations for total revenue, total expense, total inflow, total outflow, and surplus/deficit. A subroutine for the summary calculation is called with a special subroutine being used for the first month calculations where reference to a previous month is inappropriate.

Category Sums

The total value is set to zero. A loop steps through the given range of items adding each month value to the summmation line and the item year total. When the loop is exited the monthly value of the summation line is added to its year total.

Summary Calculation

The first month calculation is similar to other months except that the accumulated term deposit and accumulated operating loan are specified at 5000 and 0 respectively. The minimum cash balance is specified at 500. Accumulated term deposit is set equal to current month surplus/deficit plus previous month accumulated term deposit plus previous month accumlated operating loan. If current month accumulated term deposit is negative, then current accumulated operating loan is set to the positive of this negative value and current month accumlated term deposit is set to zero. Otherwise, current accumulated term deposit retains its positive value and accumulated operating loan is set to zero. Changes to accumulated term deposit and accumulated operating loan are the difference between current month and previous month. Year totals for these items are also updated.

APPENDIX C

Program Variables

All variables are defined as integers unless redefined within the code using the symbols "$" for character strings, "!" for single precision values, or "#" for double precision values.

I,J,K are used generally as loop counters.

TRUE, FALSE - boolean values.

OK - boolean value used to indicate that a valid key has been pressed.

VERT, HOR - refer to screen line and screen column position.

ITEM - refers to a code number for the current piece of data in a cashflow statement having a descriptive label and possibly monthly, quarterly, or yearly values associated with it.

FIRST, LAST - code numbers designating a range of items.

STATUS - indicates the current level at which the user is working.

FORMAT$ - set up for displaying a series of values on a line.

Keyboard input variables used to indicate keys pressed:

BACKSPACE, ENTER, ESCAPE, NUMERIC, DEL (delete), ALPHA

CURSOR.UP, CURSOR.DOWN, CURSOR.RIGHT, CURSOR.LEFT, PAGE.UP

PAGE.DOWN, PAGE.RIGHT, PAGE.LEFT, HOME, END.FILE,

SORT.FUNCTION.KEYS

DONE, DONE.ALL, DONE.FLOWS, DONE.DATA.CHOICE, etc. these variables are used to exit user input loops usually by pressing function key, F4. However, DONE.ALL is set to true by pressing function key, F10.

Farm Financial Flows Figure

REDO is set true by pressing function key, F5 and re-starts the presentation from the beginning.

HI1, HI2 are used to indicate that a message is to be highlighted.

Data Choice Exercises

ITEM$() - an array containing item description labels.

VALU!() - an array of quarterly values for each item.

COMMENT$() - an array of instructor comments.

KODE() - an array for item code numbers which is manipulated to arrange the evaluation

display.

MATCH() - an array used to indicate the matching or lack of matching between user and correct choices of data.

Data Choice - Exercise 1

CODE() - an array of code numbers which represents the selection of items presented to the user in random order.

CODE.BEGIN() - an array of code numbers, which is a copy of CODE(), used to restore the orginal selection and carry out the evaluation.

YEAR!() - An array of calculated year totals corresponding to the selected items.

PLACE() - an array of column positions used to locate the cursor when review is called.

MOVE$ - the character string for the line to which move is to take place.

MOVE.TO - a line number to which an item is to be moved.

LEFT.COL - the left most cursor position for numeric input in moving a line.

CHOICES - indicates the number of choices to be made from a given range of items in making up a selection.

EMPTY - indicates empty lines in a selection resulting from deleting items.

KODE.TOP - indicates the top of the code number sequence in the array KODE().

Data Choice - Exercise 2

EMPTY1, EMPTY2 - indicates empty lines for page 1 and 2 resulting from the deletion of lines.

LIN - indicates the line number of the current item.

PAGE - indicates the current page.

Liquidity Analysis

RIT.VALU!() - an array of correct values.

VALU!() - an array of values subject to user input.

MON - the current month of the value at the cursor.

MON1 - the first month to be displayed on a page.

ROW - the current screen line of the cursor.

COL - the current screen column position of the cursor.

ROW1 - the first screen line for displaying a category of items.

DISPLAY - a boolean value indicating that the cursor is to be on or off.

TITLE.HEAD$ - title character string.

PART.HEAD$ - heading for a category of items.

MON.HEAD$ - heading over six months.

Data Entry

QTR.HEAD$ - heading for months used in evaluation.

QTR - indicates the current quarter of months to be displayed.

Analyses

CHOICE$() - an array of options.

CHOICE - indicates the user choice among options.

SET - indicates one of four clusters of three items displayed on a page in evaluation.

RIT.VALU!() - an array of correct values.

APPENDIX D

Program Code

Program Code Index:

	Statement #
Main Program	1000
Data Choice	6000
Farm Financial Flows Figure	9000
Exercises	19000
Exercise 1	20000
Exercise 2	25000
Common Subroutines	29000
Liquidity Analysis	30000
Data Entry	30100
Analyses	35000
Financing	36000
Price sensitivity	3XX00
Yield sensitivity	3XX00
Interest Rates	3XX00
Sales timing	3XX00
Common Subroutines	39000
Common Subroutines	50000

The following example of the program code is taken from Exercise 1.

```
1000 '****************************
1010 '* CASHFLOW LEARNING PROGRAM *
1020 '****************************
1030 GOSUB 5000 'Begin
1040 GOSUB 19000 'Data Choice - Exercises
1050 IF NOT DONE.ALL THEN IF STATUS=4 THEN LOAD "ENTRY",R
                         ELSE LOAD "ANALYSES",R
1070 GOSUB 5200 'End
1080 '              ##################################
5000 'Begin Learning Program
5010 '----------------------
5030 DEFINT A-Z:OPTION BASE 1:RANDOMIZE(VAL(RIGHT$(TIME$,2)))
5040 I=0: J=0: K=0: TRUE=-1: FALSE=0: VERT=0: ITEM=0: R1=0: R2=0:
   C1=0: C2=0
5050 I$="":MESSAGE1$="":MESSAGE2$="PRESS ANY KEY to continue":
   TYPE$="": MOVE$"": FUNCTION.KEY$=""
5080 OPEN "B:STATUS" FOR INPUT AS #1: INPUT #1, STATUS: CLOSE
5090 RETURN
5095 '              ##################################
5200 'End Cashflow Program
5210 '--------------------
5220 CLS: OPEN "b:status" FOR OUTPUT AS #1:PRINT #1, STATUS:SYSTEM
5230 '              ##################################
20000 'Data Choice - Exercise 1
20010 '************************
20020 GOSUB 21700 'Begin
20040 GOSUB 20200 'Data selection
20050 GOSUB 20600 'Display exercise
20060 GOSUB 20800 'User input
20070 GOSUB 22500 'End
20080 RETURN
20090 '             ##################################
20200 'Data Selection - Exercise 1
20210 '--------------------------
20220 LOCATE VERT,1: PRINT CHR$(32):
    LOCATE 23,27: PRINT "Data Selection in Progress":
    CODE(1)=81: CODE(2)=62: I=3 'Total Cash Inflow & Op Loan Int
20230 CHOICES=1: FIRST=1: LAST=8: GOSUB 29000 'Non-operating inflow
20250 CHOICES=2: FIRST=9: LAST=29: GOSUB 29000 'Revenue
20260 CHOICES=1: FIRST=31: LAST=38: GOSUB 29000 'Non-op outflow
20270 IF CODE(4)=>9 AND CODE(4)=17
      THEN CHOICES=1: FIRST=39: LAST=43: GOSUB 29000
      ELSE CHOICES=1: FIRST=45: LAST=50: GOSUB 29000 'Crop * Lvst
20280 IF CODE(5)=>9 AND CODE(5)=43
      THEN CHOICES=1: FIRST=39: LAST=43: GOSUB 29000
      ELSE CHOICES=1: FIRST=45: LAST=50: GOSUB 29000 'Crop & Lvst
20290 CHOICES=2: FIRST=52: LAST=61: GOSUB 29000 'Supply & service
20300 CHOICES=1: FIRST=76: LAST=80: GOSUB 29000 'Irrelevant
20310 LIST.LENGTH=11: GOSUB 29100 'Randomize the list
20320 FOR I=1 TO 15: CODE.BEGIN(I)=CODE(I): NEXT
20330 EMPTY=0: FOR I=1 TO 13: YEAR!(I)=0: NEXT
20350 RETURN
20360 '             ##################################
20600 'Display exercise 1
```

```
20610 '-----------------
20620 CLS: LOCATE 1,28: PRINT "DATA CHOICE - EXERCISE 1"
20630 LOCATE 25,1: PRINT FUNCTION.KEY$;: VERT=3+EMPTY: LOCATE VERT,3
20640 PRINT "#" TAB(13) "CASHFLOW ITEMS" TAB(33) "1ST QTR" TAB(43)
    "2ND QTR" TAB(53) "3RD QTR" TAB(63) "4TH QTR" TAB(77) "YEAR"
20650 VERT=VERT+2
20660 FOR I=1+EMPTY TO 15
20670   LOCATE VERT,1,0
20680   IF CODE(I)=81 THEN PRINT USING FORMAT1$;
        I-EMTY, ITEM$(CODE(I)), VALU!(CODE(I),1),
        VALU!(DOCE(I),2), VALU!(CODE(I),3), VALU!(CODE(I),4),
        YEAR!(I)
20700   IF CODE(I)=>82 AND CODE(I)=83 THEN PRINT USING
        FORMAT2$; ITEM$(CODE(I))), VALU!(CODE(I),1), VALU!(CODE
        (I),2), VALU!(CODE(I),3), VALU!(CODE(I),4), YEAR!(I)
20710   IF CODE(I)=>84 THEN PRINT USING FORMAT3$;ITEM$(CODE(I)),
        VALU!(CODE(I),1),VALU!(CODE(I),2),VALU!(CODE(I),3),
        VALU!(CODE(I),4)
20720   IF CODE(I)=81 OR CODE(I)=82 OR CODE(I)=83
        THEN VERT=VERT+2 ELSE VERT=VERT+1
20730 NEXT
20740 ITEM=1: VERT=5+EMPTY:LOCATE VERT,1,0:COLOR 23,0:PRINT CHR$(16)
20750 COLOR 7,0: RETURN
20760 '             ##################################
20800 'User Input - exercise 1
20810 '----------------------
20820 DONE.EXERCISE.1=FALSE: OK=FALSE
20825 WHILE NOT DONE.EXERCISE.1 AND NOT DONE.ALL
20830    GOSUB 52000 'Read keyboard
20835    IF CURSOR.UP THEN OK=TRUE: GOSUB 21000: CURSOR.UP=FALSE
20840    IF CURSOR.DOWN THEN OK=TRUE:GOSUB 21100:CURSOR.DOWN=FALSE
20850    IF NUMERIC THEN GOSUB 21400: NUMERIC=FALSE: GOSUB 20600:
                 OK=TRUE
20860    IF DEL THEN GOSUB 21200: DEL=FALSE: GOSUB 20600: OK=TRUE
20870    IF SORT.FUNCTION.KEYS THEN GOSUB 20900
20880    IF OK THEN OK=FALSE
           ELSE MESSAGE1$="This key is not in use":
              GOSUB 50000: GOSUB 52400
20885 WEND
20890 RETURN
20895 '             ##################################
20900 'Sort Function Keys
20910 '-----------------
20920 IF I$=";" THEN GOSUB 24400: GOSUB 20600: OK=TRUE 'F1: Help
20930 IF I$=""THEN GOSUB 24600: GOSUB 20600: OK=TRUE 'F2
20940 IF I$=">"THEN DONE>EXERCISE.1=TRUE: OK=TRUE 'F4: Done
20945 IF I$="?" THEN GOSUB 24500: GOSUB 20600: OK=TRUE 'F5: Redo
20950 IF I$="" THEN GOSUB 20200: GOSUB 20600: OK=TRUE 'F6: Again
20960 IF I$="B" THEN GOSUB 22900: GOSUB 20600: OK=TRUE 'F8: Check
20965 IF I$="C" THEN GOSUB 22600: GOSUB 20600: OK=TRUE 'F9: Calc
20970 IF I$="D" THEN GOSUB 50300: OK=TRUE:
      IF ESCAPE THEN ESCAPE=FALSE: GOSUB 20600 'F10: End Session
20975 SORT.FUNCTION.KEYS=FALSE
20980 RETURN
```

```
20990 '             ##################################
21000 'Cursor Up
21010 '---------
21020 LOCATE VERT,1: PRINT CHR$(32);
21030 ITEM=ITEM-1: VERT=VERT-1
21040 IF ITEM=0 THEN IF NOT EVALUATE THEN ITEM=11-EMPTY: VERT=16
                      ELSE ITEM=HI-1: VERT=HI+2
21050 IF NOT EVALUATE THEN IF CODE(ITEM+EMPTY)=81 THEN VERT=VERT-1
21055 IF EVALUATE THEN IF KODE(ITEM)=81
                THEN VERT=VERT-2: ITEM=ITEM-1
21060 LOCATE VERT,1,0: COLOR 23,0: PRINT CHR$(16);: COLOR 7,0
21070 RETURN
21080 '             ##################################
21100 'Cursor Down
21110 '-----------
21120 LOCATE VERT,1: PRINT CHR$(32);
21130 ITEM=ITEM+1: VERT=VERT+1
21140 IF NOT EVALUATE AND ITEM=12-EMPTY THEN ITEM=1: VERT=5+EMPTY
21145 IF EVALUATE AND ITEM=HI THEN ITEM=1: VERT=5
21150 IF NOT EVALUATE THEN IF ITEM1 THEN IF CODE(ITEM+EMPTY-1)
                            =81 THEN VERT=VERT+1
21155 IF EVALUATE AND KODE(ITEM)=81 THEN VERT=VERT+2: ITEM=ITEM+1
21160 LOCATE VERT,1,0: COLOR 23,0: PRINT CHR$(16);: COLOR 7,0
21170 RETURN
21180 '             ##################################
21200 'Delete a line
21210 '-------------
21220 IF CODE(ITEM+EMPTY)>80 OR CODE(ITEM+EMPTY)=62
      THEN MESSAGE1$="This line cannot be deleted":
         GOSUB 50000: RETURN
21230 LOCATE VERT,1: PRINT CHR$(16): I$=" "
21240 LOCATE 23,30: PRINT "Line " ITEM " to be deleted"
21250 LOCATE 24,24,1,11,13:PRINT "Press ENTER if OK, otherwise Esc";
21260 WHILE ASC(I$)13
21270    DEF SEG=0: POKE 1050,PEEK(1052): I$="": WHILE I$="":
     I$=INKEY$: WEND
21280    IF ASC(I$)=27 THEN LOCATE 23,1: PRINT SPACE$(80): RETURN
21290    IF ASC(I$)27 AND ASC(I$)13 THEN BEEP
21300 WEND
21310 CODE(ITEM+EMPTY)=0
21320 WHILE ITEM1
21330    SWAP CODE(ITEM+EMPTY),CODE(ITEM+EMPTY-1)
21340    ITEM=ITEM-1
21350 WEND
21360 EMPTY=EMPTY+1
21370 LOCATE 23,1: PRINT SPACE$(80)
21380 RETURN
21390 '             ##################################
21400 'Move a line
21410 '-----------
21420 LOCATE VERT,1: PRINT CHR$(16)
21430 MOVE$=I$:LOCATE 23,30:PRINT "Move line" ITEM "to line " MOVE$;
21440 LEFT.COL=POS(0)-1: HOR=POS(0): LOCATE 24,9
21450 PRINT "Only backspace, number, enter, and escape keys are
```

```
        recognized";
21460 GOSUB 29200: IF ESCAPE THEN RETURN 'Keyboard input
21470 IF MOVE.TO1 OR MOVE.TO>11-EMPTY
      THEN MESSAGE1$="The given line number was invalid":
        GOSUB 5000: RETURN
21480 IF MOVE.TOITEM THEN ADD=-1 ELSE ADD=1
21490 WHILE ITEMMOVE.TO
21500   SWAP CODE(ITEM+EMPTY),CODE(ITEM+EMPTY+ADD)
21510   ITEM=ITEM+ADD
21520 WEND
21530 LOCATE 23,1,0: PRINT SPACE$(80)
21540 RETURN
21550 '             ##################################
21700 'Begin Exercise 1
21710 '----------------
21720 CLS: LOCATE 3,24: PRINT "DATA CHOICE - EXERCISE (Level 1)"
21730 GOSUB 21900 'Display objective
21740 VERT=1: EMPTY=0: FIRST=0: LAST=0:
    MOVE.TO=0: LEFT.COL=0: CHOICES=0
21750 DIM PLACE(3), CODE(15), CODE.BEGIN(15), YEAR!(13)
21760 PLACE(1)=3: PLACE(2)=32: PLACE(3)=69
21770 CODE(12)=82: CODE(13)=83: CODE(14)=84: CODE(15)=85
21780 ITEM$(81)="      Total Cash Inflow"
21790 ITEM$(82)="     Total Cash Outflow"
21800 ITEM$(83)="  Cash In Less Cash Out"
21810 ITEM$(84)="        Operating Loan"
21820 ITEM$(85)="         Cash Balance"
21840 RETURN
21850 '             ##################################
21900 'Objective Exercise 1
21910 '--------------------
21920 LOCATE 7,27: PRINT "The idea in this exercise" CHR$(13)
    CHR$(13) TAB(27) "is to SORT the given items" CHR$(13)
CHR$(13) TAB(22) "into cash INflows and cash OUTflows"
21930 LOCATE 16,25: PRINT "When you are finished sorting," CHR$(13)
CHR$(13) TAB(22) "use function key F8 to check yourself"
21960 GOSUB 50000
21970 RETURN
21980 '             ##################################
22500 'End exercise 1
22510 '--------------
22520 ERASE PLACE, CODE, CODE.BEGIN, YEAR!
22530 RETURN
22540 '             ##################################
22600 'Calculation
22610 '-----------
22615 LOCATE VERT,1: PRINT CHR$(32)
22620 LOCATE 23,35: COLOR 0,7: PRINT "CALCULATING";: COLOR 7,0
22625 FOR I=1 TO 13: YEAR!(I)=0: NEXT: SUM!=0
22630 FOR J=1 TO 4
22640   I=1+EMPTY: VALU!(81,J)=0: VALU!(82,J)=0
22650   WHILE I13
        IF J>1 AND CODE(I)=62
          THEN VALU!(CODE(I),J)=VALU!(84,J-1)*.1
```

```
22670       IF CODE(I)=81 OR CODE(I)=82
            THEN VALU!(CODE(I),J)=SUM!: SUM!=0
            ELSE SUM!=SUM!+VALU!(CODE(I),J)
22680       YEAR!(I)=YEAR!(I)+VALU!(CODE(I),J)
22690       I=I+1
22700     WEND
22770     YEAR!(12)=YEAR!(12)+VALU!(82,J)
22780     VALU!(83,J)=VALU!(81,J)-VALU!(82,J)
22790     YEAR!(13)=YEAR!(13)+VALU!(83,J)
22800     IF J=1 THEN VALU!(85,J)=0+VALU!(83,J)-0
          ELSE VALU!(85,J)=VALU!(85,J-1)+VALU!(83,J)-VALU!(84,J-1)
22810     IF VALU!(85,J)1
          THEN VALU!(84,J)=-VALU!(85,J): VALU!(85,J)=0
          ELSE VALU!(85,J)=0
22820 NEXT
22830 RETURN
22840 '
22899 '             ##################################
22900 'Evaluation - Exercise 1
22910 '-----------------------
22920 CLS: LOCATE 1,3: PRINT "DATA CHOICE - EXERCISE 1" TAB(37);:
 COLOR 0,7: PRINT "EVALUATION";: COLOR 7,0: PRINT TAB(63)"COMMENT:"
22930 LOCATE 3,9: PRINT "YOUR CHOICE" TAB(35) "CORRECT CHOICE"
22940 R1=3: R2=22: C1=56: C2=80: GOSUB 50200 'Box
22950 LOCATE 10,20: PRINT "The Evaluation" CHR$(13) CHR$(13) TAB(23)
"is being" CHR$(13) CHR$(13) TAB(24) "set up"
22960 GOSUB 23200: GOSUB 23100: PAGE=1: GOSUB 29800
22970 LOCATE 25,31: PRINT "4 DONE EVALUATION";
22980 DONE.EVALUATION=FALSE: OK=FALSE
22990 WHILE NOT DONE.EVALUATION
23000   GOSUB 52000 'Read keyboard
23010   IF CURSOR.UP THEN OK=TRUE: GOSUB 28500: CURSOR.UP=FALSE
23015   IF CURSOR.DOWN THEN OK=TRUE: GOSUB 28600: CURSOR.DOWN=FALSE
23020   IF PAGE.UP THEN OK=TRUE: PAGE=PAGE-1: GOSUB 23400:
              PAGE.UP=FALSE
23025   IF PAGE.DOWN THEN OK=TRUE: PAGE=PAGE+1: GOSUB 23400:
              PAGE.DOWN=FALSE
23030   IF ENTER THEN OK=TRUE: GOSUB 23600: ENTER=FALSE 'Comment
23040   IF SORT.FUNCTION.KEYS AND I$">"
        THEN DONE.EVALUATION=TRUE: SORT.FUNCTION.KEY$=FALSE
23085   IF OK THEN OK=FALSE
          ELSE MESSAGE1$="This key is not in use": GOSUB 50000
23090 WEND
23095 RETURN
23099 '             ##################################
23100 'Set Up Evaluation
23110 '-----------------
23120 I=1+EMPTY: K=1
23130 SUM.ITEM=81: GOSUB 29900: PAGE2.FIRST=K
23140 SUM.ITEM=82: GOSUB 29900: PAGE3.FIRST=K:
    IF K17 THEN PAGE2.FIRST=1
23150 RETURN
23160 '             ##################################
23200 'Correct order - Exercise 1
```

```
23210 '-------------------------
23215 FOR I=1 TO 50: KODE(I)=0: NEXT
23220 FOR I=1 TO 12: KODE(I+38)=CODE.BEGIN(I): NEXT
23230 KODE.TOP=39
23240 FIRST=1: LAST=29: GOSUB 29300 'Inflows
23250 FIRST=81: LAST=81: GOSUB 29300 'Total
23260 FIRST=31: LAST=62: GOSUB 29300 'Outflows
23270 FIRST=82: LAST=82: GOSUB 29300 'Total
23280 KODE.TOP=39
23290 RETURN
23295 '             ##################################
23400 'Evaluation Page
23410 '--------------
23420 IF PAGE=0 THEN PAGE=2
23430 IF PAGE=3 THEN PAGE=1
23440 IF PAGE=2 AND PAGE2.FIRST=1 THEN PAGE=1
23450 GOSUB 29800
23460 RETURN
23470 '             ##################################
23600 'Select Comment - Exercise 1
23610 '-------------------------
23620 LOCATE VERT,1: PRINT CHR$(16)
23630 IF KODE(ITEM)=>2 AND KODE(ITEM)=8
      THEN COMMENT$=COMMENT$(1)
23640 IF KODE(ITEM)=>9 AND KODE(ITEM)=17
      THEN COMMENT$=COMMENT$(2)
23650 IF KODE(ITEM)=>19 AND KODE(ITEM)=25
      THEN COMMENT$=COMMENT$(3)
23660 IF KODE(ITEM)=>27 AND KODE(ITEM)=29
      THEN COMMENT$=COMMENT$(4)
23670 IF KODE(ITEM)=>32 AND KODE(ITEM)=38
      THEN COMMENT$=COMMENT$(5)
23680 IF KODE(ITEM)=>39 AND KODE(ITEM)=43
      THEN COMMENT$=COMMENT$(6)
23690 IF KODE(ITEM)=>45 AND KODE(ITEM)=50
      THEN COMMENT$=COMMENT$(7)
23700 IF (KODE(ITEM)=>52 AND KODE(ITEM)=54)
      OR KODE(ITEM)=61 THEN COMMENT$=COMMENT$(8)
23710 IF (KODE(ITEM)=>55 AND KODE(ITEM)=60)
      OR KODE(ITEM)=62 THEN COMMENT$=COMMENT$(9)
23720 IF KODE(ITEM)=> THEN COMMENT$=COMMENT$(10)
23730 IF KODE(ITEM)=1 THEN COMMENT$=COMMENT$(1)+COMMENT$(11)
23740 IF KODE(ITEM)=31
      THEN COMMENT$=COMMENT$(5) + SPACE$(8) + "*" + SPACE$(18)
         + "*" + COMMENT$(11)
23750 GOSUB 29600 'Display comment
23760 RETURN
23770 '             ##################################
24400 'Display Help - Exercise 1
24410 '------------------------
24420 CLS: DEF SEG=&HB000: BLOAD "CHOICHLP.SCN",0
24430 GOSUB 50000
24440 RETURN
24450 '             ##################################
```

```
24500 'Restore Data
24510 '------------
24520 FOR I=1 TO 13
24530    IF I=11 THEN CODE(I)=CODE.BEGIN(I)
24540    YEAR!(I)=0
24550 NEXT
24560 FOR I=1 TO 5: FOR J=1 TO 4: CALC!(I,J)=0: NEXT J: NEXT
24570 EMPTY=0
24580 RETURN
24590 '              ##################################
24600 'Review
24610 '------
24620 LOCATE VERT,1: PRINT CHR$(32): OK=FALSE:
    LOCATE 23,4: PRINT "COMPLETED FLOWS FIGURE      FLOWS FIGURE
    with comments        OBJECTIVE";
24630 SELECT=1:HOR=3:LOCATE 23,3:COLOR 23,0:PRINT CHR$(16):COLOR 7,0
24640 WHILE NOT ENTER
24650    GOSUB 52000 'Read keyboard
24655    IF ENTER THEN OK=TRUE
24660    IF ESCAPE THEN ESCAPE=FALSE: LOCATE 23,1:
                PRINT SPACE$(159);: RETURN
24665    IF CURSOR.RIGHT THEN ADD=1: GOSUB 24800: OK=TRUE:
                  CURSOR.RIGHT=FALSE
24670    IF CURSOR.LEFT THEN ADD=-1: GOSUB 24800: OK=TRUE:
                  CURSOR.LEFT=FALSE
24675    IF OK THEN OK=FALSE
           ELSE MESSAGE1$="This key is not in use":
              GOSUB 5000: GOSUB 52400
24680 WEND
24690 ENTER=FALSE
24700 IF SELECT=1 THEN CLS: DEF SEG=&HB000: BLOAD "FLOWSFIG.SCN",0:
              GOSUB 50000
24710 IF SELECT=2 THEN LOAD "FLOWS",R
24720 IF SELECT=3 THEN CLS: LOCATE 1,28: PRINT
              "DATA CHOICE - EXERCISE 1": GOSUB 21900
24730 RETURN
24740 '              ##################################
24800 'Shift cursor right
24810 '-----------------
24820 LOCATE 23,HOR: PRINT CHR$(32)
24830 SELECT=SELECT+ADD
24840 IF SELECT=4 THEN SELECT=1: HOR=3
24850 IF SELECT=0 THEN SELECT=3: HOR=69 ELSE HOR=PLACE(SELECT)
24860 LOCATE 23,HOR: COLOR 23,0: PRINT CHR$(16): COLOR 7,0
24870 RETURN
24880 '              ##################################
29000 'Choosing routine
29010 '---------------
29020 J=I
29030 WHILE CHOICES>0
29040    VALID=TRUE: CODE(I)=INT(RND*(LAST-FIRST))+FIRST: K=I-1
29050    IF CODE(I)=18 OR CODE(I)=26 OR CODE(I)=27 OR
         CODE(I)=30 OR CODE(I)=44 OR CODE(I)=51
         THEN VALID=FALSE: K=J-1
```

```
29060   WHILE K=>J
29065     IF CODE(I)=CODE(K) THEN VALID=FALSE: K=J
29070     K=K-1
29080   WEND
29085   IF VALID THEN CHOICES=CHOICES-1: I=I+1
29090 WEND
29095 RETURN
29099 '             ##################################
29100 'Randomize list
29110 '--------------
29120 I=1
29310 WHILE ILIST.LENGTH
29140   J=INT(RND*(LIST.LENGTH-1-I))+I+1
29150   SWAP CODE(J),CODE(I): I=I+1
29160 WEND
29170 RETURN
29180 '             ##################################
29200 'Move Line Input
29210 '--------------
29220 DONE=FALSE: OK=FALSE
29230 WHILE NOT DONE
29240   LOCATE 23,HOR,1,12,13
29250   GOSUB 52000 'Read keyboard
29255   IF ENTER THEN DONE=TRUE: OK=TRUE: ENTER=FALSE
29260   IF ESCAPE THEN LOCATE 23,1,0: PRINT SPACE$(80): DONE=TRUE:
            OK=TRUE
29270   IF BACKSPACE THEN BACKSPACE=FALSE: OK=TRUE: IF POS(0)-1
        LEFT.COL-1 THEN LOCATE 23,POS(0)-1,0: PRINT
        CHR$(32);: HOR=POS(0)-1: LOCATE 23,HOR,1,12,13:
        MOVE$=LEFT$(MOVE$,LEN(MOVE$)-1)
29275   IF NUMERIC AND POS(0)=>LEFT.COL AND POS(0)
        LEFT.COL+1 THEN PRINT I$;: MOVE$=MOVE$+I$: HOR=POS(0):
      NUMERIC=FLASE: OK=TRUE
29280   IF OK THEN OK=FALSE
          ELSE MESSAGE1$="This key is not in use": GOSUB 50000
29285 WEND
29290 MOVE.TO=VAL(MOVE$): LOCATE 1,1,0
29295 RETURN
29299 '             ##################################
29300 'Category Search
29310 '--------------
29320 J=KODE.TOP
29330 WHILE J=50
29340   IF KODE(J)=>FIRST AND KODE(J)=LAST
        THEN GOSUB 29400 'Move to top
29350   J=J+1
29360 WEND
29370 RETURN
29380 '             ##################################
29400 'Bubble-up routine
29410 '----------------
29420 WHILE JKODE.TOP
29430   SWAP KODE(J),KODE(J-1)
29440   J=J-1
```

```
29450 WEND
29460 KODE.TOP=KODE.TOP+1
29470 RETURN
29480 '            ##################################
29500 'Search Correct Selection
29510 '-----------------------
29520 J=KODE.TOP
29530 WHILE NOT FOUND AND KODE(J)SUM.ITEM
29540   IF KODE(J)=CODE(I) THEN GOSUB 29400: FOUND=TRUE
29550   J=J+1
29560 WEND
29570 RETURN
29580 '            ##################################
29600 'Display Comment
29610 '--------------
29620 FIRST=1: LONG=21: LAST=21: VERTL=4
29630 WHILE MID$(COMMENT$,FIRST,1)" "
        AND MID$(COMMENT$,FIRST,1)""
29640   WHILE MID$(COMMENT$,LAST,1)" " AND MID$(COMMENT$,LAST
          ,1)" " AND MID$(COMMENT$,LAST,1)""
29650     LONG=LONG-1: LAST=LAST-1
29660   WEND
29670   LOCATE VERTL,58: PRINT MID$(COMMENT$,FIRST,LONG)
29680   IF MID$(COMMENT$,LAST,1)=" "
        THEN FIRST=LAST+1 ELSE FIRST=LAST+2
29690   IF MID$(COMMENT$,LAST+1,1)=" " THEN FIRST=LAST+2
29700   LAST=FIRST+20: LONG=21: VERTL=VERTL+1
29710 WEND
29720 GOSUB 50000
29730 FOR I=VERTL TO 4 STEP -1: LOCATE I,57,0: PRINT SPACE$(23):NEXT
29740 LOCATE VERT,1: COLOR 23,0: PRINT CHR$(16): COLOR 7,0
29750 RETURN
29760 '            ##################################
29800 'Display Evaluation Page
29805 '----------------------
29810 VERT=5: FOR I=5 TO 22: LOCATE I,1,0: PRINT SPACE$(55): NEXT
29820 IF PAGE=1
      THEN FIRST=1: IF PAGE2.FIRST=1 THEN LAST=PAGE3.FIRST-1
                       ELSE LAST=PAGE2.FIRST-1
29830 IF PAGE=2 THEN FIRST=PAGE2.FIRST: LAST=PAGE3.FIRST-1
29835 IF PAGE=3 THEN FIRST=PAGE3.FIRST:
            IF PAGE4.FIRST=PAGE3.FIRST THEN LAST=PAGE.END
                         ELSE LAST=PAGE4.FIRST-1
29840 IF PAGE=4 THEN FIRST=PAGE4.FIRST: LAST=PAGE.END
29845 FOR I=FIRST TO LAST
29850   LOCATE VERT,3
29855   IF MATCH(I)=1 THEN PRINT ITEM$(KODE(I))
29860   IF MATCH(I)=2 THEN PRINT STRING$(24,45) TAB(30)
                     ITEM$(KODE(I))
29865   IF MATCH(I)=3 THEN PRINT ITEM$(KODE(I)) TAB(30)
                     ITEM$(KODE(I))
29870   IF KODE(I)=65 OR KODE(I)=68 OR KODE(I)=81 OR KODE(I)=82
        THEN VERT=VERT+2 ELSE VERT=VERT+1
29875 NEXT
```

```
29880 ITEM=FIRST: VERT=5: IF KODE(ITEM)=65 OR KODE(ITEM)=68
                    THEN ITEM=FIRST+1: VERT=7
29885 LOCATE VERT,1,0: COLOR 23,0: PRINT CHR$(16): COLOR 7,0
29890 RETURN
29895 '             ##########################################
29900 'Build Up Evaluation Kode
29910 '-----------------------
29920 WHILE CODE(I)SUM.ITEM
29930    KODE(K)=CODE(I)
29940    FOUND=FALSE: GOSUB 29500: IF FOUND THEN MATCH(K)=3
                                   ELSE MATCH(K)=1
29950    K=K+1: I=I+1
29960 WEND
29970 J=KODE.TOP
29975 WHILE KODE(J)SUM.ITEM
29980  KODE(K)=KODE(J): MATCH(K)=2:
     KODE.TOP=KODE.TOP+1: K=K+1: J=J+1
29985 WEND
29990 KODE(K)=SUM.ITEM:MATCH(K)=3:
    KODE.TOP=KODE.TOP+1: K=K+1: I=I+1
29995 RETURN
29999 '             ##########################################
```